Interactive Visualization of Simulation Data for Geospatial Decision Support

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Dipl.-Ing. Daniel Cornel

Registration Number 0726194

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Second advisor: Dipl.-Ing. Dr.techn. Jürgen Waser

The dissertation has been reviewed by:

<table>
<tr><td>_____</td><td>_____</td></tr>
<tr><td>Natalia Andrienko</td><td>Elmar Eisemann</td></tr>
</table>

Vienna, 19th January, 2020

_____
Daniel Cornel

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Daniel Cornel
Rüdengasse 12/11, 1030 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 19. Jänner 2020

_____
Daniel Cornel

Acknowledgements

Kurzfassung

Überschwemmungen sind katastrophale Ereignisse, die jedes Jahr Tausende von Menschenleben fordern. Für die Vorhersage solcher Ereignisse sind interaktive Entscheidungshilfesysteme mit integrierter Hochwassersimulation ein unverzichtbares Werkzeug geworden. Der technische Fortschritt erlaubt nun die Simulation von Hochwasserszenarien nie dagewesener Größe und Auflösung, was zu großen Mengen zeitabhängiger Daten führt. Die Menge an Simulationsdaten wird durch den Einsatz von Ensemblesimulationen weiter erhöht, um Vorhersagen robuster zu machen, was zu riesigen hochdimensionalen und mit Unsicherheit behafteten Daten führt, die nicht mehr manuell zu bewältigen sind. Daher sind neue Strategien erforderlich, um diese Daten zu filtern und nur die wichtigsten Informationen anzuzeigen, um damit Fachexperten bei ihrer täglichen Arbeit zu unterstützen. Diese Arbeit umfasst auch die Kommunikation der Ergebnisse an Entscheidungsträger, Rettungsdienste, Interessenvertreter und die Allgemeinbevölkerung. Ein modernes Entscheidungshilfesystem muss deshalb visuelle Resultate liefern können, die für Fachexperten nützlich, aber auch für eine breitere Öffentlichkeit verständlich sind. Darüber hinaus muss der gesamte Prozess der Simulation, Analyse und Visualisierung interaktiv ablaufen, um einen effizienten Arbeitsablauf zu gewährleisten, was zu erheblichen Zeiteinschränkungen für das gesamte System führt.

In dieser Arbeit stellen wir neuartige Visualisierungstechniken für zeitabhängige und mit Unsicherheit behaftete Hochwasser-, Logistik- und Fußgängersimulationsdaten für ein interaktives Entscheidungshilfesystem vor. Da die vielfältigen Aufgaben im Hochwassermanagement sehr unterschiedliche Visualisierungen für unterschiedliche Zielgruppen erfordern, bieten wir Lösungen für Schlüsselaufgaben in Form von aufgabenspezifischen und benutzerspezifischen Visualisierungen. So kann der Benutzer bei Bedarf detaillierte Informationen ein- oder ausblenden, um verständliche und ästhetische Visualisierungen zur Unterstützung der jeweiligen Aufgabe zu erhalten. Um die Auswirkungen von Hochwasserereignissen auf ein einzelnes Gebäude zu ermitteln, ist nur eine kleine Teilmenge aller verfügbaren Daten relevant, weshalb wir eine Lösung vorschlagen, um diese Informationen aus den massiven Simulationsdaten zu isolieren. Um die inhärente Unsicherheit der daraus resultierenden Vorhersagen von Schäden und Gefahren zu kommunizieren, führen wir einen einheitlichen Stil zur Visualisierung der Unsicherheit im geografischen Kontext ein. Statt der direkten, zeitabhängigen Darstellung von Simulationsdaten schlagen wir weiter die Verwendung von bidirektionalen Flussdiagrammen mit mehreren Komponenten

als vereinfachte Darstellung beliebiger Materialflüsse vor. Für die verständliche Kommunikation von Hochwasserrisiken kann jedoch die direkte, zeitabhängige Darstellung von Simulationsdaten auch gewünscht sein. Neben den offensichtlichen Herausforderungen durch die komplexen Simulationsdaten führt die Diskretisierung der Daten zu zusätzlichen Problemen für die realistische Visualisierung von Wasseroberflächen, für die wir robuste, für Echtzeitanwendungen geeignete Lösungen vorschlagen. Alle unsere Ergebnisse wurden durch eine kontinuierliche Zusammenarbeit mit Fachexperten aus verschiedenen Arbeitsbereichen mit Hochwasserbezug gewonnen. Die gründliche Evaluierung unserer Arbeit durch diese Experten bestätigt die Relevanz und den Nutzen unserer vorgestellten Lösungen.

Abstract

Floods are catastrophic events that claim thousands of human lives every year. For the prediction of these events, interactive decision support systems with integrated flood simulation have become a vital tool. Recent technological advances made it possible to simulate flooding scenarios of unprecedented scale and resolution, resulting in very large time-dependent data. The amount of simulation data is further amplified by the use of ensemble simulations to make predictions more robust, yielding high-dimensional and uncertain data far too large for manual exploration. New strategies are therefore needed to filter these data and to display only the most important information to support domain experts in their daily work. This includes the communication of results to decision makers, emergency services, stakeholders, and the general public. A modern decision support system has to be able to provide visual results that are useful for domain experts, but also comprehensible for larger audiences. Furthermore, for an efficient workflow, the entire process of simulation, analysis, and visualization has to happen in an interactive fashion, putting serious time constraints on the system.

In this thesis, we present novel visualization techniques for time-dependent and uncertain flood, logistics, and pedestrian simulation data for an interactive decision support system. As the heterogeneous tasks in flood management require very diverse visualizations for different target audiences, we provide solutions to key tasks in the form of task-specific and user-specific visualizations. This allows the user to show or hide detailed information on demand to obtain comprehensible and aesthetic visualizations to support the task at hand. In order to identify the impact of flooding incidents on a building of interest, only a small subset of all available data is relevant, which is why we propose a solution to isolate this information from the massive simulation data. To communicate the inherent uncertainty of resulting predictions of damages and hazards, we introduce a consistent style for visualizing the uncertainty within the geospatial context. Instead of directly showing simulation data in a time-dependent manner, we propose the use of bidirectional flow maps with multiple components as a simplified representation of arbitrary material flows. For the communication of flood risks in a comprehensible way, however, the direct visualization of simulation data over time can be desired. Apart from the obvious challenges of the complex simulation data, the discrete nature of the data introduces additional problems for the realistic visualization of water surfaces, for which we propose robust solutions suitable for real-time applications. All of our findings have been acquired through a continuous collaboration with domain experts from several flood-related fields

of work. The thorough evaluation of our work by these experts confirms the relevance and usefulness of our presented solutions.

Contents

Introduction

1.1 Motivation

When a flood occurs, lives are in danger. Over the last two decades, more than 3000 flood disasters have been recorded [WHM18], making floods the most frequent natural disasters worldwide. An estimated two billion people were directly affected by these disasters, causing many fatalities and high economic damage. Floods also present a great danger to critical infrastructure on which we rely, such as water and power supplies, medical care, and transport networks. Damage to industrial and chemical plants can result in long-lasting environmental disasters, damage to the sewage system can spread waterborne diseases. In short, flood disasters are catastrophic events that severely affect our life in countless aspects. Unfortunately, both frequency and intensity of flood disasters have increased in the recent past [BHP$^+$17] and are likely to increase even more in the future [FBS$^+$12].

A flood is defined by the Intergovernmental Panel on Climate Change as the *overflowing of the normal confines of a stream or other body of water, or the accumulation of water over areas that are not normally submerged* [FBS$^+$12]. The main causes of floods are long-lasting precipitation, intense storms, snow melting, tsunamis, and dam breaks. This thesis puts an emphasis on flooding incidents that frequently occur in urban and rural regions, largely excluding coastal flooding. In particular, we focus on fluvial and pluvial floods, i.e., river floods and heavy rains, which we review in the following.

River floods and overtopping   If a river carries more water than usual, its water level rises until it overtops its banks. As cities have traditionally been built in close proximity of rivers, river floods have been accompanying people for millennia. With data from historical floods, river flooding scenarios can be predicted quite reliably. In high-risk areas, long-term mitigation measures such as dikes and floodwalls can be deployed, which are intended to protect citizens and infrastructure from frequent flooding. However, as

Figure 1.1: Examples of river flooding incidents. (a) Flooding by floodwall overtopping. (b) Flooding by dike breach.

Copyright: (a) THOMAS FREY / EPA / picturedesk.com. (b) JENS WOLF / EPA / picturedesk.com.

these protection structures are only effective up to a certain water level, overtopping can still occur during extreme flooding events, as illustrated in Figure 1.1a.

Dike and floodwall breaches   A second, devastating flooding incident is the structural failure of flood protection due to breaches, as illustrated in Figure 1.1b. Dike and floodwall breaches can be caused, for example, by debris or trees carried by the water, or by wave impact. Dikes in particular are also prone to erosion, infiltration, and seepage, which

Figure 1.2: Examples of common flooding incidents. (a) Heavy rain. (b) Sewer overflow.

Copyright: (a) Helge Treichel / dpa / picturedesk.com. (b) Corinna John, CC BY-NC-SA 2.0, flickr.com/photos/binarycoco/9379272016, last visited on January, 18th 2020.

can weaken their foundation [D'E07]. Such structural failures are almost impossible to predict, and since dikes and floodwalls are designed as primary flood protection, there are usually no backup protection measures in case of failure.

Heavy rain Heavy rain causes floods if the surface water cannot infiltrate the soil or be taken up by storm drains quickly enough and runs off the surface, as illustrated in Figure 1.2a. The impact of heavy rain events is further amplified by paved surfaces, which inhibit infiltration and increase surface runoff. In recent years, flooding by heavy

rains has received increasing attention by public authorities and the scientific community because of the increasing frequency of such events and the damages they cause. The observed increase of extreme precipitation events can at least partially be linked to anthropogenic influences on the climate [FBS$^+$12]. By their nature, heavy rains are hard to predict and affect a large region at the same time, which makes the implementation of local, long-term mitigation measures an increasingly important task.

Sewer overflows During precipitation and flooding events, surface water is constantly being taken up by storm drains and ideally transported away by the sewer network. If, however, the sewer network itself is overburdened or congested, the hydrostatic pressure in the pipes can push water out of the system at the lower-lying areas of the network and cause sewer overflows, as illustrated in Figure 1.2b. The danger of these overflows lies in the ostensible incoherence with surface flooding in other regions, which can lead to flooding in areas that are otherwise protected against surface floods.

1.1.1 Flood Management

The primary goal of flood management authorities is to mitigate the risks of the aforementioned flooding incidents to protect people and infrastructure. The large range of responsibilities of flood management is usually subdivided into prevention, preparation, response, and recovery tasks [Lum07], of which prevention and preparation are of particular importance for this thesis.

Flood simulation Planning and decision making in the flood prevention and preparation phases are based on predictions of the severity, location, and timing of expected floodings. For the prediction of flooding behavior, flooding incidents are usually approximated by computer simulation. A comprehensive overview of simulation methods is provided by Teng et al. [TJV$^+$17]. The most common ones use a hydrodynamic model and numerically solve the shallow water equations with the finite-element method. These simulations work on a discrete abstraction of the real world, including terrain elevations, building walls, and flood protection structures. For this discretization, the two popular methods are unstructured triangular meshes and rectangular grids. Flood simulation on triangular meshes [AB05, HLSH13, LMMG15] is highly flexible with respect to approximating complex boundaries such as building walls or protection barriers. Compared to regular grids, however, it is relatively slow, which makes it unfavorable for large-scale scenarios. A study by Leskens et al. [LBHS14] suggests that the high runtime of flood simulations was a main factor for their neglect in flood management in the past. Shallow water simulation on rectangular grids [KP07, HWP$^+$15, BHN$^+$19] provides a high performance, because the grid structure is well suited for the GPU data model. This enables an efficient parallelization of the simulation [HPW$^+$16] to drastically reduce computation times, which is why grid-based flood simulation using the finite-volume method is the state of the art in interactive flood management. However, as also pointed out by Leskens et al. [LBHS14], simulation results have often been met with scepticism

Figure 1.3: Examples of long-term flood protection measures. (a) Retention areas. (b) Dikes. (c) Floodwalls.

Figure 1.4: Examples of short-term flood protection measures. (a) Sandbags. (b) AquaRiwa. (c) Aquabarrier.

Copyright: (a) Andre Baumann, flickr.com/photos/schneller2000/123725798, last visited on January, 18th 2020. (b) AQUARIWA GmbH, aquariwa.de/die-einsaetze/aquariwa-in-koeln-januar-2012/, last visited on January, 18th 2020. (c) Aliq, public domain, commons.wikimedia.org/wiki/File:Waterbarri%C3%A8re_geodesign_aqua_barrier_(5).JPG, last visited on January, 18th 2020.

by decision makers in the past because of the complexity and the questioned reliability of the flood models. In order to increase the acceptance of flood simulation, these concerns should be met with a systematic validation of flood models with analytical and historical flood data, as provided by Horváth et al. [HBK+20], for example.

Ensemble simulations However, even a small variation in one of the simulation parameters can lead to a very different outcome in terms of water propagation and resulting damages. It is therefore impossible to accurately predict a flooding event. Yet, when varying the parameters over multiple predictions, some of the outcomes might be more likely than others. Thus, the aggregation of multiple predictions in *ensemble simulations* makes it possible to derive probabilities of the predicted outcomes. These uncertain flood hazards are combined with the potential flood damages to an uncertain flood risk, which is the basis for informed decision making in flood management. Recent advances in graphics hardware—to which the grid-based simulation is tailored—make it possible to cover hundreds or thousands of possible flooding scenarios, making predictions more reliable. This has been demonstrated, for example, in flood forecasting [CP09], flood protection planning [RWF+13], the automatic creation of response plans [WKS+14], and the estimated impact of severe weather events on infrastructure [KZX+14].

Flood mitigation Long before floods occur, flood managers seek to reduce the damage potential in flood-endangered areas based on the estimated flood risk. For example, flood management authorities in many countries—including all member states of the European Union [Eur07]—are obliged to issue risk maps for river flooding of varying severity. Area and building precaution measures are taken to avoid building projects in high-risk areas and to make existing structures more resilient. This also means to sensitize the general public to flood risks and communicate the conduct in case of emergency. To reduce the flood risk in such areas, long-term mitigation measures such as flood control structures are designed and deployed. These include, for example, retention areas, dikes, and floodwalls, as illustrated in Figure 1.3. At the same time, action plans for the response phase during flooding events are created and practiced, for example for the deployment of short-term protection measures, such as mobile flood barriers. There are several types of flood barriers to choose from, which come with different benefits and limitations that have to be weighed. For example, sandbag barriers are sturdy and flexible in construction, inexpensive, and usually readily available, but have a high construction time requiring a lot of construction personnel. Also, a pyramid-like stacking as visible in Figure 1.4a is needed for stability, which requires a lot of material and limits their effective height. AquaRiwa barriers illustrated in Figure 1.4b consist of plastic tanks filled with water, making them easily deployable in the response phase, but increasingly unstable with a rising water level. Aquabarriers illustrated in Figure 1.4c are constructed with a watertight plane over a metal frame, stabilized by the pressure of the water they withhold. This design makes them very robust and easy to deploy, but relatively expensive compared to the other options. As all barriers have to be deployed quickly in the response phase, detailed action plans for their deployment have to be
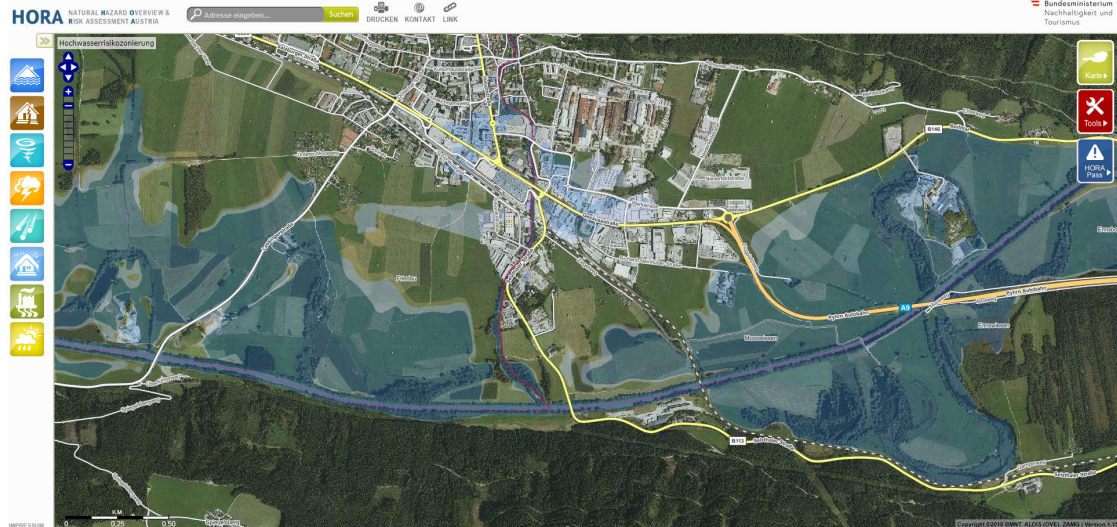
Figure 1.5: Example of a public flood risk information service for river flooding of different severity in Austria.

Copyright: BMNT, ALDIS (OVE), ZAMG, own image retrieved from hora.gv.at, last visited on January, 18th 2020.

developed. Such action plans do not only include construction plans, but also estimated construction times and the logistics of moving the construction material from depots to the construction sites. For different flooding scenarios, different action plans for different levels of protection or different areas need to be readily available. During the response phase, i.e., when a flood occurs, flood managers coordinate the execution of the action plans with the help of emergency services.

1.1.2 Decision Support

Decision making in flood management is hard, because each decision has real-world consequences that affect people's lives. When minimizing the flood risk within the limited response time and budget, some risk always remains. Therefore, decision makers have to prioritize mitigation measures. Lives have to be protected before property, and some infrastructure, buildings, and regions might be considered more important to protect than others. These include hospitals, retirement homes, pharmacies, schools, zoos, cultural heritage sites, public transport networks, and many others. In case of severe flooding, this means to accept damage to private property in order to ensure the protection of areas deemed more important. Such decisions have to be well substantiated and justified and the reasons leading to these decisions have to be communicated clearly to stakeholders and the general public. Public communication is generally an important aspect of flood management. Due to the severity of flooding events, it is important that everyone involved understands the provided information and correctly assesses the situation. This

Figure 1.6: Monitoring of a sketched retention area in 3Di.

Copyright: 3Di Water Management/Nelen & Schuurmans, vimeo.com/318215338, last visited on January, 18th 2020.

is why flooding information is usually provided to the general public in the familiar format of interactive 2D maps. A characteristic example of publicly available flood risk information systems is shown in Figure 1.5. Decisions are often not made by flood managers themselves, but by higher authorities they answer to. In this case, all relevant facts to arrive at a decision also have to be clearly communicated to the decision makers, who are usually no experts in the field of flood management.

As decision making is hard, software tools are used to assist in the process of arriving at a decision. There exists a variety of established flood simulation tools for engineering that have been used in flood management, such as TUFLOW [TUF], RiverFlow2D [Riv], or MIKE FLOOD [MIK]. These tools do not offer any functionality to visualize the simulation results, but provide interfaces to common geographic information system (GIS) software and rely on their visualization capabilities. This is an often time-consuming process, which imposes the task of visualization on GIS experts. Even worse, the results of a study by Leskens et al. [LBHS14] show that during flood response phases in the past, the simulated water propagation alone was not considered useful by decision makers at all. Instead, they usually relied on severely simplified approximations for deciding on actions. According to Leskens et al., this was partly due to the time it took to visualize the simulation results. To the other part, it was due to a mismatch in the information the flood models could provide and the information the decision makers actually required. This includes, for example, the predicted consequences of mitigation measures and clear statements about which areas to evacuate. This is why modern decision support systems already interpret the results of flood simulations and then
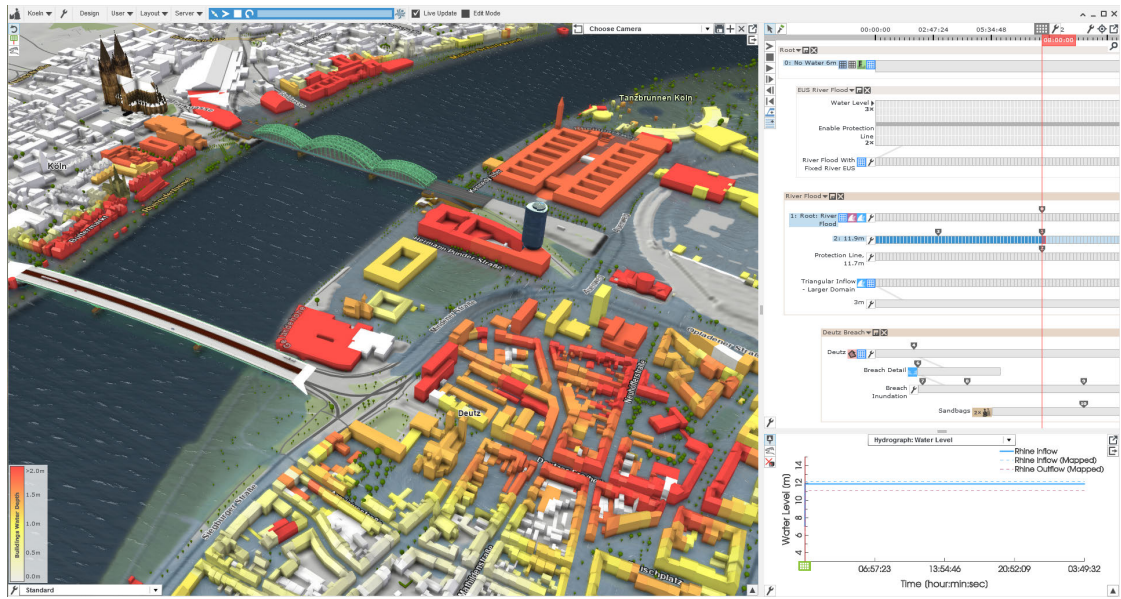
Figure 1.7: Estimated building damages during river flooding in Visdom [Vis].

provide meaningful conclusions through integrated visualization. As an example for such a system with integrated simulation, analysis, and visualization, the 3Di system [3Di] for water management is shown in Figure 1.6. The *Visdom* system [Vis] shown in Figure 1.7 serves as a framework for the solutions developed in this thesis and is described in more detail below. The main goal of these tools is to process and filter the massive amounts of uncertain and time-dependent data, extract the most important information, and present it in a comprehensible way.

A current trend in the presentation of this information is the embedding in virtual 3D environments [WFR+10, Flo, KdH13, NS13, ARMN16, VGB+16, LKT+17] or even augmented reality [SD18, HHL18] as opposed to traditional 2D maps known from GIS tools. The main benefit of 3D visualization—in particular when applying realistic surface rendering—is the familiarity of the viewer with three-dimensionality from the real world as well as the prevalence of this kind of visualization in entertainment software. It is generally assumed that this helps with the interpretability of the visualization and the orientation and navigation in the virtual environment. This assumption is supported by the results of a user study among flood management personnel conducted by Leskens et al. [LKT+17], which suggest that 3D visualization is better suited for understanding the severity and consequences of flooding scenarios than 2D visualization. Grottel et al. [GSH+15] have received similar feedback by non-experts who called their 3D visualization intuitive and useful for presentation and discussion.

Visdom [Vis] is a scenario-based decision support system that combines simulation, analysis, 2D and 3D visualization in one tool. This avoids the need to involve model specialists and visualization experts for the interpretation and presentation of results, thus
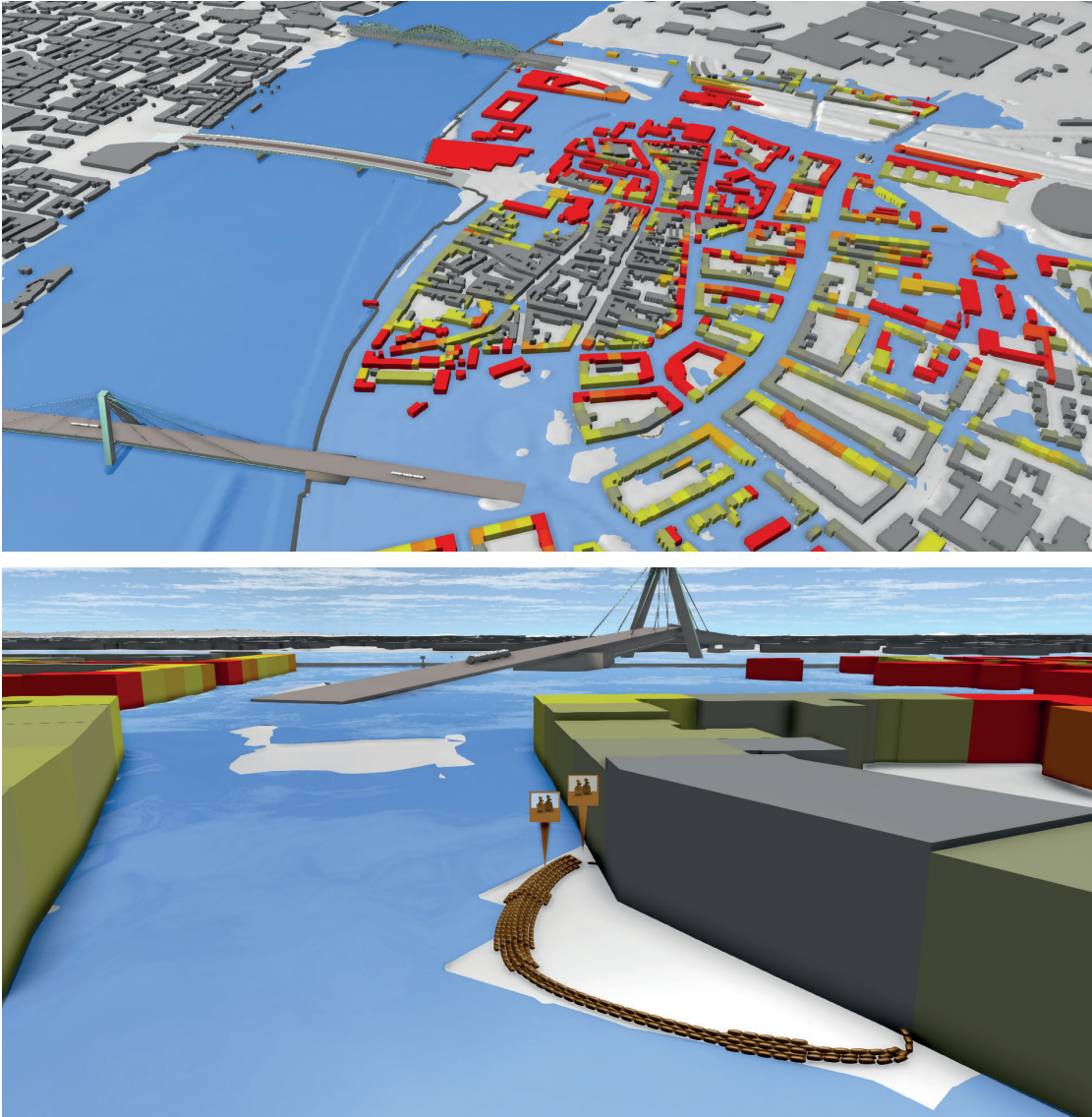
Figure 1.8: 3D visualization capabilities of Visdom at the beginning of this thesis.

streamlining the decision-making process. The generic and modular design of Visdom enables the use of a multitude of different simulations, for example for surface flooding, rainfall runoff, and underground water in sewers, as well as agent-based logistics and evacuation. The implemented *world lines* concept [WFR⁺10] allows the user to quickly create and switch between scenarios, which are basically alternative realities that can be compared. This concept also directly extends to ensemble simulations for uncertain predictions [WKS⁺14]. As with other decision support systems, analyzing and visualizing the resulting time-dependent and uncertain data in an interactive manner is a tremendous challenge due to the high-dimensionality and large scale of the data. This challenge is the

driving force for the development of novel analysis and visualization techniques presented in this thesis. The existing Visdom system serves as the basis for all proposed solutions. The current visualization capabilities of Visdom are shown in Figure 1.7. The state of the 3D visualization at the beginning of this thesis in late 2014 is illustrated in Figure 1.8.

1.2 Problem Statement

Interactive decision support in flood management is an increasingly important topic that is driven by the advances in flood modeling and the increasing capabilities of graphics hardware used for simulation. The efficient implementation of flood models already makes it possible to simulate flooding incidents many times faster than real time. This trend shifts the limits of feasibility ever further and allows for much more detailed and robust predictions. For example, the spatial resolution of simulations can be increased, while at the same time increasing the scale of the simulation from cities to an entire country. This already results in vast amounts of time-dependent simulation data that need to be processed and visualized, which further multiply to unmanageably large, high-dimensional data with the use of ensemble simulations. It is now easier than ever to become completely overwhelmed by data. We have long passed the point where it was possible to manually investigate every simulation result in detail, so novel strategies for the automatic processing of simulation data have to assist in this task.

As reported by Leskens et al. [LBHS14], decision makers often do not even want to investigate raw simulation data, but demand conclusions and recommendations for further actions. This requires not only an immediate analysis of the data, but also comprehensible and expressive visualizations that highlight the most important information. While simulation and analysis benefit significantly from recent technological advances, visualization does not, because visualizing more information does not usually lead to better visualizations. Instead, novel and intelligent visualization techniques are needed that selectively show the information that is most relevant to the viewer.

For example, current decision support systems can estimate and visualize flood risks in an area of interest, which helps flood managers with the planning of new protection measures. However, it may be assumed that homeowners are primarily interested in the vulnerability of their own property. In a tool designed for the public, instead of visualizing the impact of a single flood on every building, a more expedient approach for homeowners would be to visualize the impact of every flood on a single building. This would allow them to investigate which flooding incidents their home is most vulnerable to and how to protect against them. Therefore, visualizations have to become *user-specific* and *task-specific*, which they are currently not.

Visualizations should always be as simple and expressive as possible, which is especially true if they are intended for the general public. Currently, flood risk information is most commonly provided in the form of 2D maps. It is not always easy to read such maps, and the use of abstract visualization techniques such as glyphs or legend-based coloring makes them even harder to comprehend. However, as of the beginning of this thesis in

late 2014, there has not been much effort to visualize the data more realistically in the familiar 3D environment. As it is difficult to convey the uncertainty of simulation data in an image understandable by non-experts, public information systems usually hide this information altogether. Even for experts, the visualization of uncertain geospatial data is an open problem.

In summary, there is a lack of visualization techniques that meet the requirements of the evolving field of interactive decision support to cope with large-scale, high-dimensional, time-dependent, uncertain data in real time. Simulation data continuously grow in size and complexity, but intelligent algorithms to process, filter, and comprehensibly display these data on the fly for specific tasks and users are missing.

1.3 Aim of the Work

The overarching objective of this thesis is the improvement and extension of decision support systems in flood management in order to increase their widespread acceptance and, ultimately, make our habitats more resilient to flooding events. Recent technological advances have made it possible to simulate such events with great accuracy, but existing software does not go much further than visualizing raw simulation data. As these data have become nearly unmanageable, decision support systems have to assist in their interpretation and provide conclusions to solve the specific tasks at hand. In this thesis, we provide a variety of task-specific visualization techniques intended for such systems as an interactive abstraction layer between the data and the users. The main responsibility of these techniques is the isolation and display of important information from the data, where the importance of information depends on the task specified by the user interactively.

As communication is a key aspect of flood management, information always has to be presented in a way that the target audience can understand. For our proposed solutions, we consider very diverse user groups, including flood modeling experts, hydrologists, flood managers, logistics experts, civil engineers, emergency service personnel, insurers, politicians, and members of the general public. These user groups have very different levels of knowledge in the fields of flood modeling and visualization, but at the same time also different tasks to fulfill. It seems utopian to fit all of these requirements into one universally comprehensible visualization. Instead, our goal is to develop a modular visualization system in which details can be added interactively to fit the users' needs and expertise. A fairly straightforward visualization would be the embedding of simulation results in the 3D environment as a blue water surface overlaying the terrain. More abstract visualizations such as flow paths or indicators of uncertainty can then be added on demand. Based on the assumption that visualizations in the familiar 3D environment are more immersive and accessible than traditional 2D techniques, we put a strong emphasis on 3D visualization in this thesis. However, as a lot of experts feel most comfortable solving certain tasks in 2D views known from GIS tools, we have to enable these traditional workflows as well. Hence, our goal is to provide visualizations that work

well in both 2D and 3D views and allow for seamless switching between the views.

From a technical point of view, our goal of the fully interactive processing of flood simulation data has serious implications for the visualization pipeline. We have to create a real-time rendering system capable of processing dynamic data in the order of several gigabytes, for which novel data structures and algorithms tailored to the modern GPU architecture for efficient parallelization need to be developed. Apart from the required performance of the system, it also has to produce aesthetic visualizations without visual artifacts to increase the viewer's trust in the presented information. With this goal in mind, we seek to develop strategies to embed abstract information in the three-dimensional geospatial domain in an intuitive way with the help of visual metaphors. Here, we need to tackle the additional challenges of visualization in three dimensions, such as occlusions and perspective changes. Many similar challenges have already been addressed in the field of real-time rendering, which is why we seek to extend already existing techniques to fit our needs. Regarding the nature of the data, there is a distinctive difference between the artistic real-time rendering for entertainment purposes and the real-time visualization for scientific purposes. A lot of established real-time rendering techniques used in video games, for example, have been devised for use cases similar to ours, but for much smaller and crafted data. One goal of this thesis is therefore the adaption and optimization of existing rendering techniques to our application to benefit from recent accomplishments in the large research field of real-time rendering.

In summary, this work considers the new challenges of decision support in flood management and aims at developing techniques to

- extract the information relevant for a specific task from a large set of time-dependent, multidimensional simulation data,

- visualize the relevant information and its inherent uncertainty in an aesthetic and functional way,

- provide user-specific visualizations on demand to increase comprehensibility among different user groups,

- exploit the latest advances in graphics hardware to achieve full interactivity of processing and visualization in an interactive decision support system.

1.4   Outline and Individual Contributions

The main body of this thesis is organized in three chapters, in which visualization solutions for specific tasks within the field of flood management are presented. Each of these chapters is based on a peer-reviewed paper published in the *Computer Graphics Forum* journal [CKS+15, CKS+16, CBK+19]. In the following, we provide a short summary of each paper. While the author of this thesis is the first author of all of these papers, they are the result of productive collaborations of several researchers. As the principal

rendering engine developer for Visdom, the author's main focus and work is the design and efficient implementation of visualization techniques, but also computational geometry approaches. In order to point out the author's further individual contributions in the considered papers, we give a short overview of each co-author's contributions. The considered papers are:
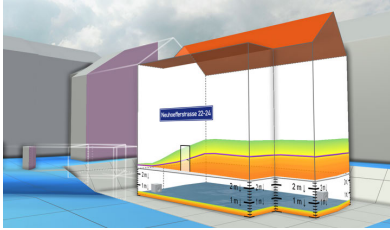
Paper 1: Daniel Cornel, Artem Konev, Bernhard Sadransky, Zsolt Horváth, Eduard Gröller, and Jürgen Waser. Visualization of Object-Centered Vulnerability to Possible Flood Hazards. *Computer Graphics Forum (Proceedings EuroVis 2015)*, 34(3):331–341, 2015. **Best Paper Award (3rd Place).**
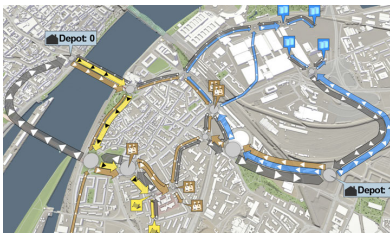
Paper 2: Daniel Cornel, Artem Konev, Bernhard Sadransky, Zsolt Horváth, Andrea Brambilla, Ivan Viola, and Jürgen Waser. Composite Flow Maps. *Computer Graphics Forum (Proceedings EuroVis 2016)*, 35(3):461–470, 2016.

Paper 3: Daniel Cornel, Andreas Buttinger-Kreuzhuber, Artem Konev, Zsolt Horváth, Michael Wimmer, Raimund Heidrich, and Jürgen Waser. Interactive Visualization of Flood and Heavy Rain Simulations. *Computer Graphics Forum (Proceedings EuroVis 2019)*, 38(3):25–39, 2019. **Best Paper Award.**

1.4.1 Paper 1: Visualization of Object-Centered Vulnerability to Possible Flood Hazards

Summary   As flooding events tend to happen more frequently, there is a growing demand for understanding the vulnerability of infrastructure to flood-related hazards. Such demand exists both for flood management personnel and the general public. Modern software tools are capable of generating uncertainty-aware flood predictions. These data can be used to understand the potential vulnerability of individual objects to disaster events. However, the information addressing individual objects is incomplete, scattered, and hard to extract. In Chapter 2, we address the vulnerability to flood-related hazards focusing on an individual, important building. Our approach is based on the automatic extraction of relevant information from a large collection of pre-simulated flooding events, called a scenario pool. From this pool, we generate uncertainty-aware visualizations conveying the vulnerability of the building of interest to different kinds of flooding events. On the one hand, we display the adverse effects of the disaster on a detailed level, ranging from damage inflicted on the building façades or cellars to the accessibility of the important infrastructure in the vicinity. On the other hand, we provide visual indications

of the events to which the building of interest is vulnerable in particular. Our visual encodings are displayed in the context of urban 3D renderings to establish an intuitive relation between geospatial and abstract information. We combine all the visualizations in a lightweight interface that enables the user to study the impacts and vulnerabilities of interest and explore the scenarios of choice. We evaluated our solution with experts involved in flood management and public communication. For animated results, we refer to the video accompanying the original paper [Vul].

Individual contributions   The first author **Daniel Cornel** devised and realized the embedding of vulnerability and impact visualizations in the geospatial domain with a consistent visual style called the uncertainty pattern in the paper. The visualization of uncertain vulnerabilities to flooding includes interactive vulnerability profiles along floodwalls and around sewer locations. The visualization of uncertain flooding impacts includes the indicators for accessibility along street lines and building polygons as well as the interactive façade plots for flooding probabilities. The author also conducted the evaluation of results with domain experts and wrote the main part of the paper. **Artem Konev** was involved in the writing of the paper and contributed to the preparation of data for visualization, which includes a model for cellar flooding and the estimation of street and building accessibility. **Bernhard Sadransky** extended the user interface of Visdom to support multiple user models and realized the interactive 2D visualization of the vulnerability to floodwall overtoppings and heavy rains. **Zsolt Horváth** extended the flood simulation to improve the performance of the required ensemble simulations and helped with the creation of paper figures. **Eduard Gröller** assisted in the creation of a paper plan, devised the idea of adaptive cutaways for focus objects, and provided a thorough review of the paper manuscript. **Jürgen Waser** supervised and coordinated the development of the paper, helped in writing, created most of the figures and the accompanying video, and prepared the presented use cases. He was responsible for the aggregation of simulation data over multiple scenarios. Together with the first author, he also conducted the evaluation of results with domain experts.

1.4.2   Paper 2: Composite Flow Maps

Summary   Flow maps are widely used to provide an overview of geospatial transportation data. Existing solutions lack the support for the interactive exploration of multiple flow components at once. Flow components are given by different materials being transported, different flow directions, or by the need for comparing alternative scenarios. In Chapter 3, we combine flows as individual ribbons in one composite flow map. The presented approach can handle an arbitrary number of sources and sinks. To avoid visual clutter, we simplify our flow maps based on a force-driven algorithm, accounting for restrictions with respect to application semantics. The goal is to preserve important characteristics of the geospatial context. This feature also enables us to highlight relevant spatial information on top of the flow map such as traffic conditions or accessibility. The flow map is computed on the basis of flows between zones. We describe a method for

automatically deriving zones from geospatial data according to application requirements. We demonstrate the method in real-world applications, including transportation logistics, evacuation procedures, and water simulation. Our results were evaluated with experts from corresponding fields. For animated results, we refer to the video accompanying the original paper [Com].

Individual contributions The first author **Daniel Cornel** was responsible for most of the scientific contributions. He developed the force-driven flow map layout algorithm with multiple bidirectional components based on a graph representation of material flow between zones. This includes overlay visualizations of context-related data and a mechanism for zoom-dependent and importance-driven continuous levels of detail. He also conducted the evaluation of results with domain experts, wrote the main part of the paper, and created the accompanying video. **Artem Konev** devised the zonation algorithm and the representation of material flow as a directed acyclic graph. His tasks included the processing of geospatial and routing data gathered from open services and the unification of data across different use cases for the layout algorithm. He also contributed to the writing of the paper. **Bernhard Sadransky** realized the coupling of the novel 3D visualization technique with the 2D user interface to enable dynamic legend updates and two-way interaction. **Zsolt Horváth** created the many schematic figures in the paper. **Andrea Brambilla** provided the initial idea and prototype of visualizing material flow (in computational fluid dynamics data) between zones with arrows. He also wrote the main portion of the related work section. Together with Andrea Brambilla, **Ivan Viola** provided the initial paper idea of a zone-based visualization of material transport. He assisted in the creation of a paper plan and provided a thorough review of the paper manuscript. **Jürgen Waser** supervised and coordinated the development of the paper, helped in writing, and created most of the figures. He applied the visualization technique to pedestrian simulation and uncertain flooding data and prepared the use cases presented in the paper. Together with the first author, he conducted the evaluation of the results with domain experts. He was also responsible for the processing of simulation data and for the material distribution analysis.

1.4.3 Paper 3: Interactive Visualization of Flood and Heavy Rain Simulations

Summary In Chapter 4, we present a real-time technique to visualize large-scale adaptive height fields with C^1-continuous surface reconstruction. Grid-based shallow water simulation is an indispensable tool for interactive flood management applications that require fast predictions of possible disasters. As simulations become increasingly powerful, height fields defined on adaptive grids are often the only viable option to store and process the massive simulation data. Visualizing these simulation results requires the reconstruction of a continuous surface from the spatially discrete simulation data. For regular grids, fast linear and cubic interpolations are commonly used for surface reconstruction. For adaptive grids, however, there exists no higher-order interpolation

technique fast enough for interactive applications. Our proposed technique bridges the gap between fast linear and expensive higher-order interpolation for adaptive surface reconstruction. During reconstruction, no matter if regular or adaptive, discretization and interpolation artifacts can occur, which domain experts consider misleading and unaesthetic. We take into account boundary conditions to eliminate these artifacts, which include water climbing uphill, diving towards walls, and leaking through thin objects. The reconstructed height fields are visualized with continuous levels of detail using view-dependent recursive tessellation. We apply realistic water shading with visual cues for better depth perception and add waves and foam synthesized from the simulation data to emphasize flow directions. The versatility and performance of our technique are demonstrated in various real-world scenarios. A survey conducted with domain experts of different backgrounds and concerned citizens proves the usefulness and effectiveness of our technique. For animated results, we refer to the video accompanying the original paper [Vid].

Individual contributions   The first author **Daniel Cornel** coordinated the development of the paper and was responsible for all scientific contributions. In particular, he devised the adaptive grid interpolation for C^1-continuous surface reconstruction, the strategies to remove visual artifacts in water height fields, and the foam and wave synthesis from simulation data. For these solutions, he also developed the crucial GPU implementation, which was in part made available along with the paper. He wrote most of the paper, conducted an evaluation of the results with domain experts, and prepared the questionnaire for the online survey. **Andreas Buttinger-Kreuzhuber** provided the formal description of the adaptive grid interpolation and the handling of transition regions and helped in writing the paper. He also assisted in the formulation of wave functions for wave synthesis that can be evaluated efficiently on the GPU. **Artem Konev** helped with writing the paper, in particular the introduction and overview sections. **Zsolt Horváth** created all schematic figures in the paper. **Michael Wimmer** provided several thorough reviews of the paper and helped with the multiple revisions. **Raimund Heidrich** helped with the coordination of the online survey and distributed the questionnaire among domain experts. **Jürgen Waser** supervised the development of the paper and helped in writing. He prepared all demonstrated use cases and created most of the paper figures along with the accompanying video. He also created the videos used in the online survey and helped in conducting the evaluation of the results with domain experts.

1.4.4   Related Co-Authored Publications

During the course of this thesis from late 2014 to 2020, the author also co-authored a number of other publications in the fields of visualization and flood modeling that are not part of this thesis. Most of these publications use preliminary or final versions of the visualization techniques discussed in this thesis to support the presentation of their contributions. This demonstrates the robustness of the discussed techniques and their applicability to a wide range of use cases.
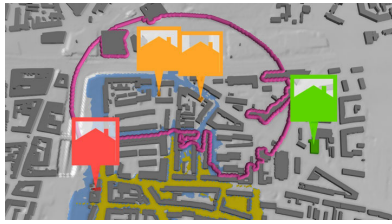
Artem Konev, Jürgen Waser, Bernhard Sadransky, **Daniel Cornel**, Rui A. P. Perdigão, Zsolt Horváth, and M. Eduard Gröller. Run Watchers: Automatic Simulation-Based Decision Support in Flood Management. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1873–1882, 2014.
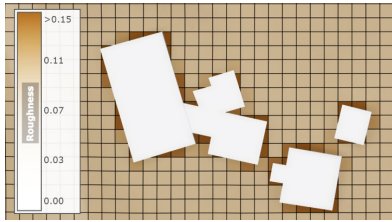
Zsolt Horváth, Jürgen Waser, **Daniel Cornel**, Artem Konev, and Günter Blöschl. Validation of the HWP14 Scheme for the Shallow Water Equations on Real-World Cases in Stockerau, Austria. 6th *Annual Symposium of the Doctoral Programme on Water Resource Systems*, Poster, 2015.
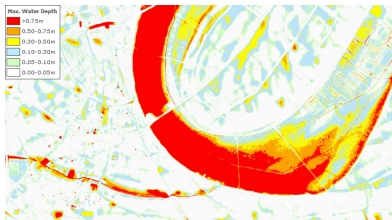
Zsolt Horváth, Jürgen Waser, Andreas Buttinger-Kreuzhuber, **Daniel Cornel**, Artem Konev, and Günter Blöschl. Comparison and Validation of Three Shallow Water Schemes on Synthetic and Real-World Cases. 7th *Annual Symposium of the Doctoral Programme on Water Resource Systems*, Poster, 2016.

Zsolt Horváth, Rui A. P. Perdigão, Jürgen Waser, **Daniel Cornel**, Artem Konev, and Günter Blöschl. Kepler Shuffle for Real-World Flood Simulations on GPUs. *The International Journal of High Performance Computing Applications*, 30(4):379–395, 2016.
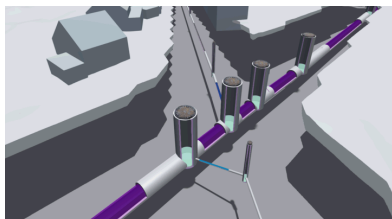
Artem Konev, Manuel Matusich, Ivan Viola, Hendrik Schulze, **Daniel Cornel**, and Jürgen Waser. Fast Cutaway Visualization of Sub-Terrain Tubular Networks. *Computers & Graphics*, 75:25–35, 2018.
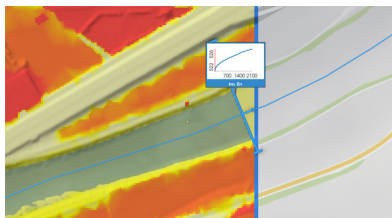
Andreas Buttinger-Kreuzhuber, Jürgen Waser, Zsolt Horváth, **Daniel Cornel**, Artem Konev, and Günter Blöschl. Outflow Boundaries for Hydrodynamic Simulations at Ungauged Locations. 9th *Annual Symposium of the Doctoral Programme on Water Resource Systems*, Poster, 2018.

Jürgen Waser, Artem Konev, and **Daniel Cornel**. On-the-Fly Decision Support in Flood Management. *GIM International*, 32(6):22–25, 2018.
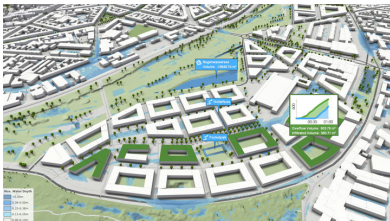
Jürgen Waser, Ingo Schwerdorf, Andreas Buttinger-Kreuzhuber, **Daniel Cornel**, Hendrik Schulze, and Günter Blöschl. Interaktive Simulationen als Entscheidungshilfe für wassersensible Stadtgestaltung. *Tag der Hydrologie*, Poster, 2019.
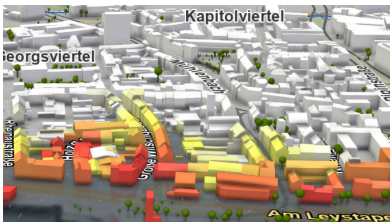
Silvana Zechmeister, **Daniel Cornel**, and Jürgen Waser. 3D Annotations for Geospatial Decision Support Systems. *Journal of WSCG*, 27(2):141–150, 2019.
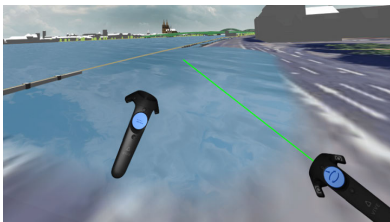
Katharina Krösl, Harald Steinlechner, Johanna Donabauer, **Daniel Cornel**, and Jürgen Waser. Master of Disaster: Virtual-Reality Response Training in Disaster Management. *Proceedings of the 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry*, 49:1–2, 2019.
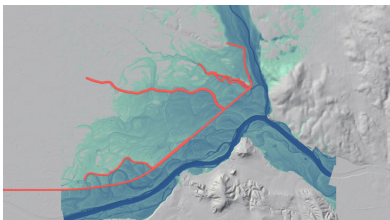
Zsolt Horváth, Andreas Buttinger-Kreuzhuber, Artem Konev, **Daniel Cornel**, Jürgen Komma, Günter Blöschl, Sebastian Noelle, and Jürgen Waser. Comparison of Fast Shallow-Water Schemes on Real-World Floods. *Journal of Hydraulic Engineering*, 146(1):05019005:1–16, 2020.

Object-Centered Vulnerability

2.1 Introduction

As elaborated in Section 1.1, recent climate studies suggest that natural disasters such as floods are likely to happen more often in the future. Measures are taken to make population and infrastructure less vulnerable to these threats, for which flood simulations are becoming increasingly important. Ensemble simulations are frequently used to make predictions more reliable, resulting in vast amounts of heterogeneous data addressing multiple aspects of the hazard and its impact on the domain of interest. Visual analytics approaches and tools come to the human's aid mitigating this data complexity. However, it is still a challenging and tedious task even for a technically-skilled domain expert to extract the information relevant for particular objects. Flood managers may need such information to understand the vulnerability of some important infrastructure, for example a hospital, to possible flood hazards. On the other hand, for individuals of the general public, the vulnerability of their personal "habitat" may be of interest. This can be the safety of their homes, the ability of their children to attend school, or the accessibility of a hospital or their favorite grocery store.

Even having an interactive decision support tool at hand, a flood manager would need to spend many hours on isolating the required information from the whole lot of data output by the simulation. To our knowledge, no solution exists that performs such information extraction, nor do the available tools support the subsequent integration of such information into a convenient representation. For non-expert users, the situation is even more complicated. Unfortunately, the information relevant to them is usually scattered among multiple heterogeneous sources and/or incomplete. Currently, a person would first need to study flood-related brochures, then visit a dedicated web page to consult flood risk maps (see Figure 1.5). For most areas, such maps are available for river flooding only and offer data for a very coarse sampling of water levels. After finding the relevant buildings and learning which range of water levels might affect them, the

Figure 2.1: Understanding the uncertain vulnerability of a selected building to a multitude of flooding scenarios. (a) Adverse impact on a selected building, including cellar flooding. The probability of water reaching a particular level varies around the building and is mapped onto the façades. (b) Vulnerability of the building to floodwall breaches, shown along the floodwall.

person would possibly need to spend even more time checking online maps and routing services for connections and trying to bring all the aspects together in their mind. Yet he or she would not be able to learn any further details about the expected impact on the building of interest, for example the exposure of particular façades to the flood water (see Figure 2.1a). Moreover, no publicly available services consider heavy rains, sewer overflows, or levee breaches, and convey the uncertainty behind the conclusions drawn.

In this chapter, we present a system for assessing and visualizing flood-related vulnerability and impacts focusing on a particular object. We call these *object-centered vulnerability* and *object-centered impacts*, respectively. Our approach is based on precomputing a large pool of possible incident scenarios with ensemble simulations. Using the data from the pool, we, on the one hand, generate uncertainty-aware visualizations conveying the vulnerability of the object of interest to possible flood hazards. We combine 2D and 3D visualization to display the vulnerability over the incident space. From this, the user can understand, for example, what water levels affect the building of interest or which breach locations are particularly dangerous to it (see Figure 2.1b). On the other hand, we create object-centered visualizations of aggregated impact using familiar visual metaphors. With the presented approach, instead of manually exploring thousands of flooding scenarios, the user just picks an address and studies the visualizations automatically created by the system.

The presented visualization techniques target two major user groups, namely flood management experts and the general public. For experts, it is important to fully understand the risks of possible incidents to mitigate them by countermeasures. They need to interactively explore the different scenarios and identify vulnerable regions with the inherent uncertainty. Non-experts often have no deep knowledge of flood management and just want clear and simple answers to whether their home and belongings could be in danger. Thus, our solution is user-specific and differs in the level of detail employed when presenting the uncertainty.

In summary, the scientific contributions presented in this chapter are:

- Automatic extraction of object-centered impact and vulnerability from a large pool of pre-simulated scenarios

- Uncertainty-aware visualizations of fine-grained impacts on a building and the accessibility of important infrastructure with respect to it

- Visualizations of uncertain vulnerability to flood-related hazards, mapped onto the geospatial representation of hazard parameters

- Selection of scenarios by means of direct interaction with the presented visualizations, avoiding the need for abstract navigation tools

2.2 Related Work

Simulations, where a process development is modeled over time, have long been a standard tool for studying real-world phenomena [COT$^+$11]. However, such modeling is plagued with uncertainty originating from imperfect initial conditions, model incompleteness, or intrinsic stochasticity of the modeled processes [ODR$^+$02]. One way to handle such uncertainty is to use ensemble simulations, where multiple simulations are conducted using slightly different initial conditions or even different models [PSH$^+$05]. For example, Finney et al. [FGM$^+$11] present an ensemble modeling approach for wildfire propagation taking into account different weather conditions. Booshehrian et al. [BMPM12] utilize ensemble simulations to support decision makers in fisheries management. Fischer et al. [FFGS14] propose a method to evaluate the quality of snow avalanche simulation ensembles involved in planning protection measures.

One of the largest applications for ensemble simulations is in climate modeling and weather predictions [GR05, Col07, SATS07, SSB$^+$09, Par10]. Hulme et al. [HBA$^+$99] investigate the effects of human-induced long-term climate change relative to natural climate variability. Krishnamurti et al. [KKZ$^+$00] compare the performance of a multi-model ensemble forecast against all used individual models. Dai et al. [DMW$^+$01] attempt to predict the climate change over the twenty-first century with respect to two major scenarios of carbon dioxide emission. Tailor and Buizza [TB03] use weather ensemble predictions to forecast electricity demand. Komma et al. [KRBH07] examine the propagation of the ensemble distribution of precipitation forecasts in the catchment system. Blöschl et al. [BRK08] present a model for flash flood prediction. Cloke and Pappenberger [CP09] review the trends and challenges in flood forecasting. Demeritt et al. [DNCP10] discuss the issues of communication and use of ensemble flood forecasts. Ribičić et al. [RWF$^+$13] simulate ensembles of flooding scenarios for protection planning in urban areas. Waser et al. [WKS$^+$14] provide a scalable extension of this approach to multiple ensemble dimensions to create a large pool of flooding scenarios and response plans. VASA [KZX$^+$14] is a tool for interactive computational steering that combines multiple simulations in a single pipeline. Further works on disaster management include the FLIWAS [GWL07] flood information and management system and the SECOM [SEC] serious game. To our knowledge, no system computes and visualizes natural catastrophe-related data focused on a particular object or set of objects.

Creating visualizations of flood-related data suitable for a wider audience requires a careful selection of techniques. In this regard, dedicated online resources and brochures can be of interest [Köl]. A cutaway technique can be used to indicate indoor flooding [Lün]. Current evacuation status, evacuation zones, and accessible shelters can be conveniently shown together in an online application [Pin]. In visualization literature, Maas and Döllner present object-integrated annotations and labels [MD06, MD08]. Lorenz and Döllner [LD10] provide techniques to map surface property data on 3D objects. Cutaway techniques are described for geological modeling [LHV13], medical data [VKG04], or generic polygonal scenes [BF08]. An efficient implementation of stylized lines for outline rendering is described in the work of Rougier [Rou13]. A proper indication of uncertainty is required

for the derived data [BOL12]. MacEachren et al. [MRH$^+$05] review the uncertainty visualization agenda for geospatial data. Mirzargar et al. [MWK14] suggest a method for summarizing ensembles of 2D and 3D curves. Correll and Gleicher [CG14] vote for a cautious use of error bars for the 2D visualization of uncertain data and suggest different approaches, including gradient-based ones.

2.3 Scenario Pool and Uncertainty Treatment

In this chapter, we focus on visualizing object-centered impacts and vulnerabilities with respect to flood-related hazards. We call an *impact* the damage inflicted by the flood water upon the buildings of interest, or the inaccessibility of important locations, for example hospitals or schools, due to inundation. By *vulnerability* we mean the degree of being exposed to flood-related hazards.

The cornerstone of the approach is the *scenario pool*. This is essentially a large database of pre-simulated flooding scenarios [WKS$^+$14], which we created by using the shallow water 2D flood simulation engine integrated in Visdom [HWP$^+$15]. For our case study in Cologne, Germany, the pool maintains scenarios for the four basic types of flood-related incidents illustrated in Figure 1.1 and Figure 1.2, which are floodwall overtoppings, floodwall breaches, heavy rains, and sewer overflows. For each incident type, we varied some of its fundamental characteristics, thus obtaining four multidimensional ensembles of flooding events. For floodwall overtoppings, we picked ten possible water levels and ten overtopping durations. For breach events, we simulated 30 possible breach positions and five possible breach widths against four different water levels and four breach event durations. Regarding heavy rains, ten different precipitation rates were simulated for ten possible event durations. Finally, 15 alternative locations were picked to model sewer overflow events of five possible durations. Summing it up, the considered scenario pool contains 2675 different flooding scenarios.

The ensemble dimensions listed above can be divided into two groups. The dimensions of the first group (overtopping water levels, breach positions, precipitation rates, sewer locations) are used to visually map object-centered vulnerability. For example, in our visualizations, the vulnerability to a breach occurring at a particular position is shown exactly at that position along the actual floodwall (see Figure 2.1b). The other ensemble dimensions are used to treat uncertainty. For instance, for sewer overflows, modeling multiple overflow durations per sewer position increases the fidelity of the computed uncertain vulnerability. In addition, some of these ensemble dimensions are used to give more details on the presented vulnerability (for example breach widths in Figure 2.1b).

Our system provides the user with several basic exploration options. First, the user can select the incident type to be considered. From the scenario pool perspective, this means switching between the available multidimensional ensembles. Second, the user can select what he or she wants to see in the auto-generated visualizations. The two alternatives are vulnerability and impact. For each of them the user has to further specify what exactly he or she is interested in, for example vulnerability with respect to the
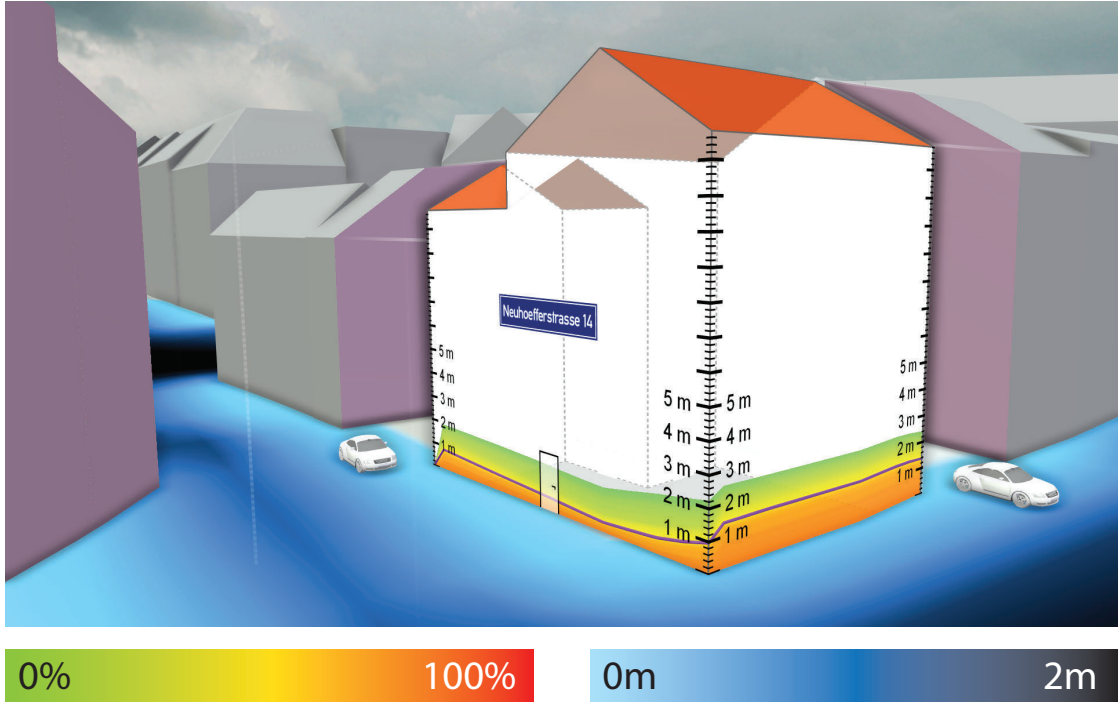
Figure 2.2: Impact visualization with façade area plots. The color indicates the probability of that particular part of the façade being exposed to water. The current water depths in the vicinity are mapped to shades of blue visualized on the terrain.

inaccessibility of hospitals. This defines what kind of information has to be extracted from the ensemble of interest and how exactly this information has to be presented. Finally, the user can pick any scenario for manual exploration. This can be done directly from the generated visualizations. For further details on the user interface, we refer to the video accompanying the original paper [Vul].

2.4   Object-Centered Impacts

To visualize the impact of flood-related hazards on a building of interest, we map the water levels aggregated over all relevant scenarios onto the building in the 3D city model. More specifically, we accumulate the façade areas exposed to water in each scenario in a density plot. The density then indicates the probability distribution for such an exposure. After applying a color transfer function, the resulting plots are displayed on the corresponding façades of the building of interest. We call them façade area plots. Example screenshots are shown in Figure 2.1a and Figure 2.2. This continuous visualization along the façades of the building allows for an easy perception of the detailed possible damages and conveys the underlying uncertainty. Additionally, the user can select a scenario by picking a water level directly on a building façade. The water level corresponding to the
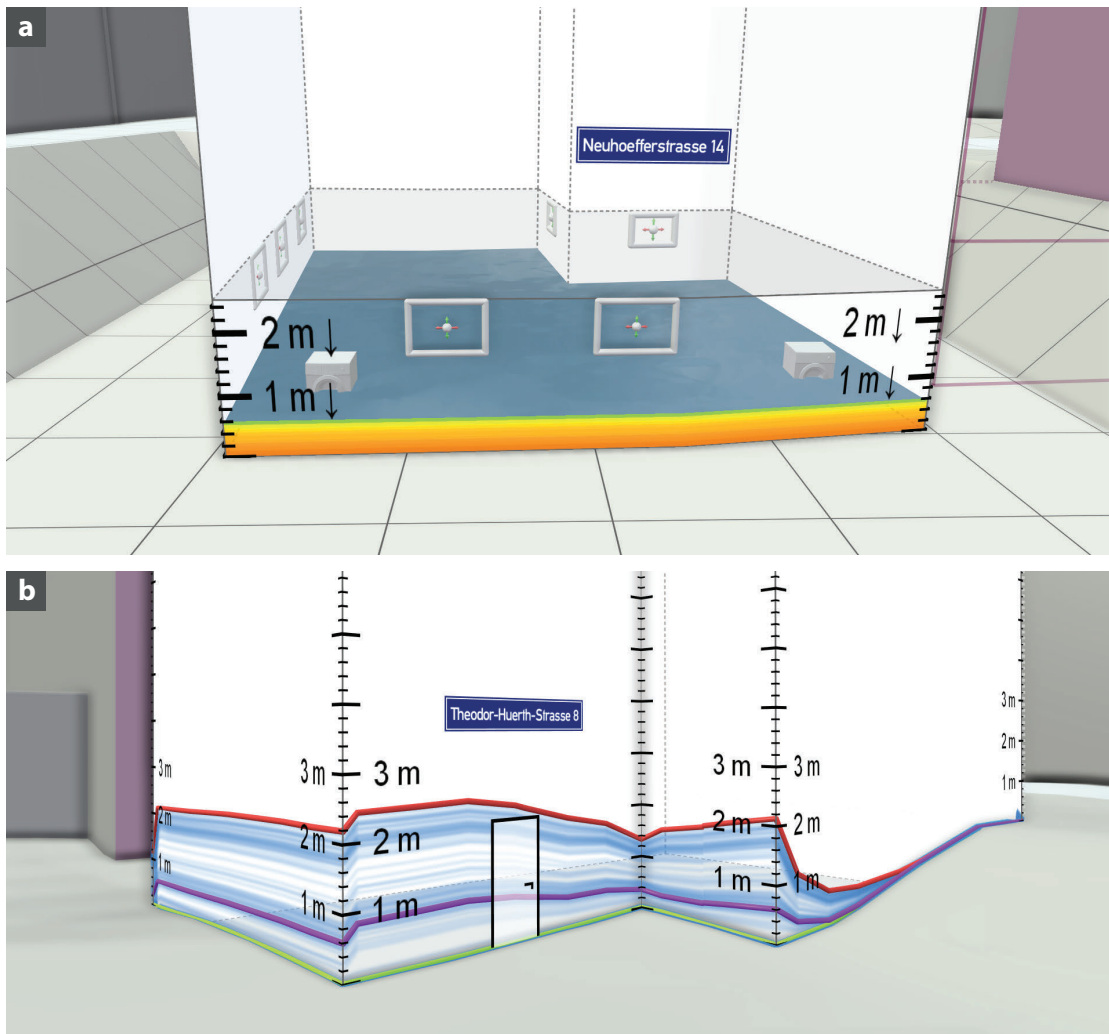
Figure 2.3: Internal and external flooding impact. (a) Estimated cellar flooding through user-sketched windows. Water levels are visualized with a façade area plot. (b) Façade line plot showing the water level probabilities for the external flooding.

selected scenario is then indicated with a purple line along the façades, and the respective water depths are shown as overlay on the terrain using shades of blue.

To improve the readability of the water levels, zoom-dependent gauges are provided on the façade canvas (see Figure 2.2). Additionally, reference objects of well-known size are given for better size comparison. A (fictitious) door is shown on the camera-facing façade so that it is always visible. At the contour edges of the building, a car is displayed such that occlusions of the façade area plots are avoided. The building of interest itself is visually emphasized by using wall and roof colors different from the neighboring buildings, and by accentuating the building outlines. The address of the building is displayed in

a billboard of a fixed screen-space size. This billboard is either displayed as a floating label above the building or, if it does not occlude the building too much, on the most prominent façade [MD08].

To avoid occlusions from neighboring buildings, adaptive cutaways [BF08] are used. We found that using a proxy cuboid as the cutaway volume rather than the building itself results in simpler and more comprehensible cut surfaces. Instead of calculating the view-dependent cutaway surfaces based on a Chamfer distance of the depth buffer, we generate them in a geometry shader from the contours extracted from the cutaway volume. In Figure 2.1a and Figure 2.2, the cut surfaces are colored in purple. Ghost lines are displayed in light gray to indicate the former shape of the cut buildings. Note that, in Figure 2.2, the viewpoint is located inside a neighboring building, which is cut away completely to allow the user a clear view on the building of interest from this angle.

Besides the exterior water levels, it is possible to visualize the water levels inside a building. For estimating this interior flooding, we consider the water inflow through leaky windows, which the user can sketch [RWG+12] directly on the façades, as illustrated in Figure 2.3a. From the sketched windows, user-specified inflow rate, and the exterior water levels, the development of the interior water levels over time is estimated. For visualization, the camera-facing façades are made transparent to show the cellar. As the cellar is usually occluded by terrain, a cutaway volume is used to virtually excavate the building. Inside the building, the maximum water level is displayed as a surface. Washing machines are used as reference objects for a better perception of water levels. Interior and exterior water levels can be displayed on the façades at the same time, as shown in Figure 2.1a.

One more representation of uncertain flooding impacts is shown in Figure 2.3b. A density plot displays the distribution of water levels. More opaque regions indicate a higher probability of water reaching this level. The minimal and maximal water levels are indicated in green and red, respectively. The purple line shows the water level for the current scenario picked by the user. In this representation, the distribution of water levels can be investigated in a convenient way, because the opacity of a certain water level is proportional to the probability of this water level being reached in any simulated scenario. In Figure 2.3b, for example, the opacity of the façade lines is higher in the regions around 1 m and 2 m than in between, from which follows that these water levels are more likely to occur than water levels in between.

2.5 Uncertainty Mapping

In this section, we describe a basic pattern for visualizing uncertain values ranging between zero and one (or, equivalently, between 0 % and 100 %). This pattern is primarily used to display an object's vulnerability to user-selected hazards. Additionally, we utilize it for impact visualization if considering inaccessibility of important infrastructure with respect to the building of interest.
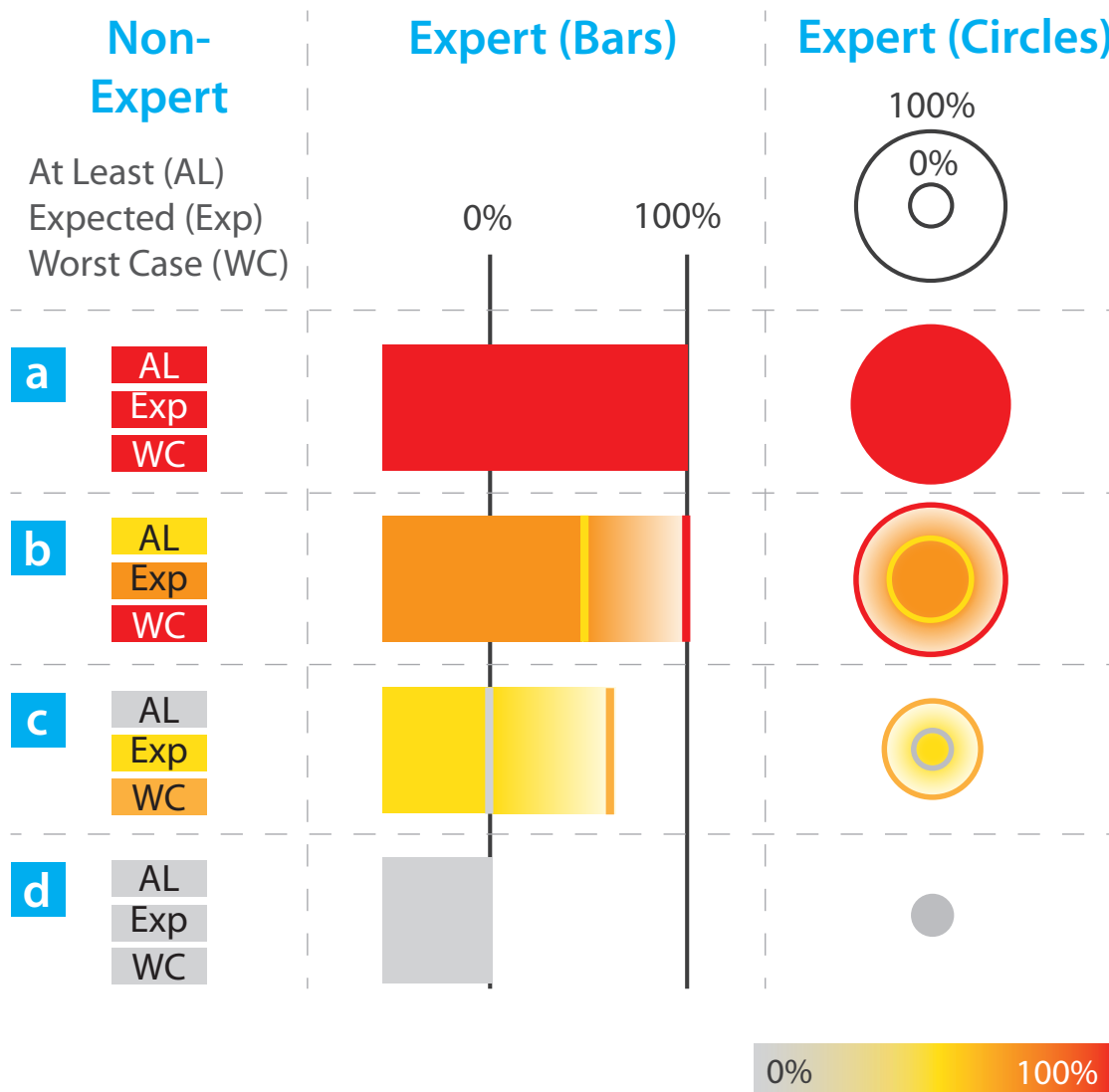
Figure 2.4: Pattern for the uncertainty-aware visualization of vulnerability. For non-experts, only one of the three choices (at least, expected, worst case) is shown at a time. For experts, this uncertainty information is combined. (a) 100 % vulnerable in all cases. (b) Vulnerable up to 100 % in some cases, at least 50 %, average 75 %. (c) Safe in some cases, but in some other cases up to 60 % vulnerable. Average vulnerability is 50 %. (d) Always safe.

The pattern is shown in Figure 2.4. We color-code the percentage according to the user-defined transfer function (bottom right). Uncertainty can be presented at two different levels of detail, thus targeting expert users, the general public, or both. For non-expert users (left column), the simplified visualization uses solid colors to present, according to the user's choice, the minimal ("at least"), average ("expected"), or maximal

("worst case") value across all scenarios under consideration. For expert users, we provide a more informative representation (middle column). The actual color encodes the average value. A context legend is provided, indicating the 0 % and 100 % reference levels with black lines. The two thick lines, colored according to the same transfer function, denote the minimal and maximal values across the considered scenarios. Between these two lines, a transparency gradient indicates the uncertainty range. Note that, regardless of the size of the gradient part, a prominent solid part conveying the average value with color is always present. The right column in Figure 2.4 shows the same pattern in a circular layout. Here, the vulnerability is proportional to the radius of the circle rather than its area, which allows us to establish the same linear scale between 0 % and 100 % as for the bars. The circular layout is convenient when visualizing vulnerability relative to a location on a map, for example, for sewer overflows.

Our design follows the well-known box plot visualization to present the uncertainty of the results in a compact and familiar way that contains the minimum, average, and maximum values. When mapped onto the spatial domain, the final visualization adopts the shape of its spatial reference object to emphasize the correspondence of abstract and geospatial data. Vulnerability to sewer overflows is displayed with a circular plot to resemble the shape of a manhole. The accessibility of a building is visualized along its ground plan and the outlines of the routes leading to it. The vulnerability to breaches is visualized along the floodwall. In all cases, the visualization is displayed to be large enough to allow for picking of a concrete scenario. The transfer functions employed for the visualization can be changed by the user via presets or by manual editing through standard techniques.

2.6   Floodwall Overtoppings and Heavy Rains

We visualize the object-centered vulnerability to floodwall overtoppings in an interactive 2D chart. Such a chart is presented in Figure 2.5, where the vulnerability is shown with respect to the damage inflicted on the building of interest. In the left part of the chart, a vertical gauge shows different water levels and the corresponding vulnerability values. The value encoding follows the convention described in Section 2.5. Note that the visualization shown in the figure is the expert version with a more detailed uncertainty indication.

The second part of the chart, displayed on the right, is organized in a table layout. The horizontal dimension of the table represents the considered overtopping event durations. Each cell visualizes the vulnerability of the scenario defined by the corresponding water level and duration. The user can pick scenarios for exploration by clicking on the cells or on the water level gauge. If a scenario is picked, the view displays a hydrograph representing the evolution of the water level in the river over time. The hydrograph is an inflow boundary condition for the simulation defined by a synthetic function with the two parameters peak water level and duration of overtopping, which are varied in the ensemble. A dedicated cursor can be used for time-navigating the scenario of interest.
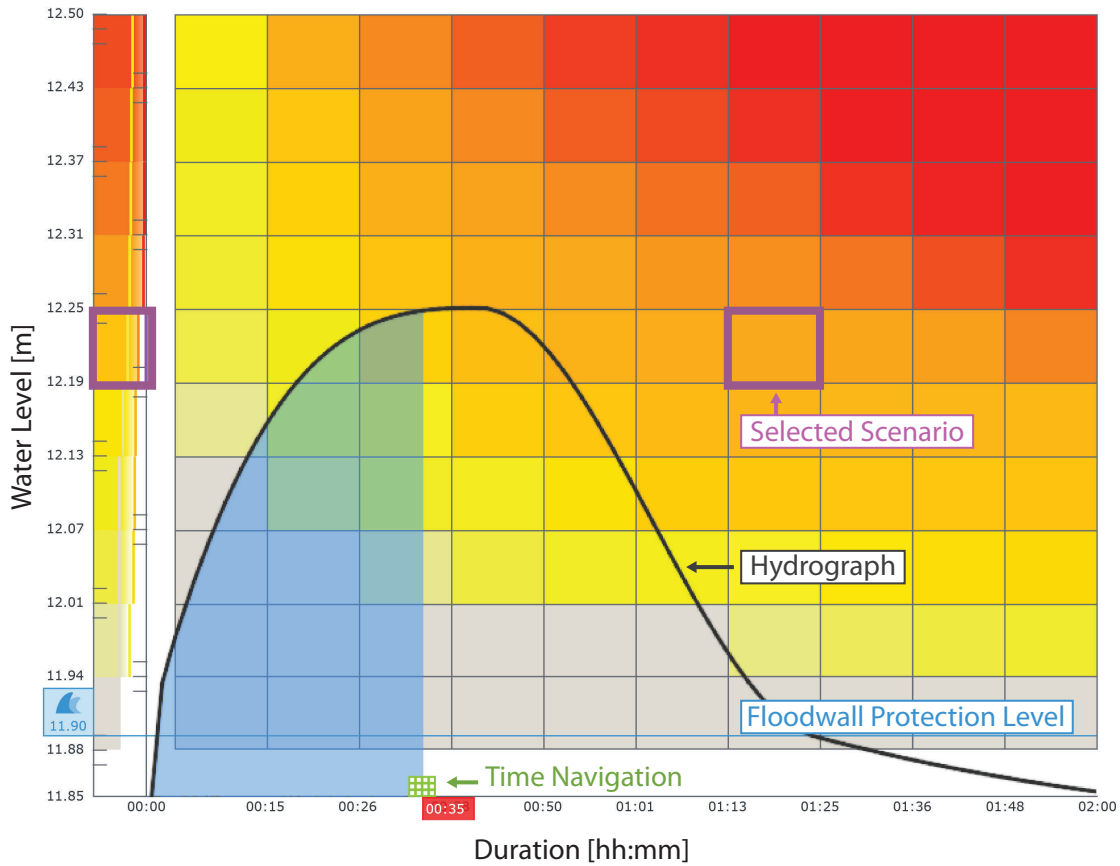
Figure 2.5: Interactive 2D chart for the vulnerability of a building to floodwall overtoppings. The gauge shows the water levels and the corresponding uncertain vulnerability values for experts. Each cell shows the vulnerability in the corresponding scenario (water level and event duration). The scenario highlighted in purple is currently selected, and the hydrograph shows the water level development for this scenario over time.

The floodwall protection level, i.e., the critical water level after which the overtopping starts (11.9 m for Cologne), is shown with a blue horizontal line. The vulnerability to heavy rains is visualized in a similar fashion. In this case, the vertical gauge indicates different precipitation rates, and the hydrograph displayed on picking a scenario shows the rain development over time.

2.7 Sewer Overflows

We visualize the vulnerability to sewer overflow events on a 3D city model, as shown in Figure 2.6. The vulnerability values are shown at the corresponding sewer locations by using the circular pattern described in Section 2.5. Note that, for non-expert users, these are circles of different radii and solid colors. In this case, both the color and the radius

Figure 2.6: Vulnerability to sewer overflows, presented for experts. (a) Sewer locations potentially dangerous for the selected building. (b) For another building, different locations are dangerous. (c) On mouse-over, a context legend is displayed, showing the 0 % and 100 % bounds. The building contours preserve the spatial context. (d) Vulnerability with respect to the accessibility of the nearest hospital.

of each circle reflect the same (user-selected) minimal, average, or maximal vulnerability across the relevant scenarios. For expert users, the gradient-based representation is used.

The user can clearly see the sewer locations potentially dangerous for the building of interest (see Figure 2.6a). For a different building selected, other sewer locations are more dangerous (see Figure 2.6b). A context legend, showing the 0 % and 100 % vulnerability bounds, is displayed on mouse-over (see Figure 2.6c). In Figure 2.6d, the vulnerability of the same building is presented with respect to the accessibility of the hospital. Two sewer locations are potentially very dangerous, and one is dangerous in the worst case only. Notice that, if some of the buildings are overlaid with the vulnerability visualization, their contours are shown to keep the context. In Figure 2.7a, the user has selected a sewer location to study the expected inundation associated with the overflow of that particular sewer. It is also possible to apply a satellite image fetched from an online service (for example from Google Maps) as a texture to provide more context and assist orientation in the scene. Different perspectives can be used for the visualizations (see Figure 2.7b).
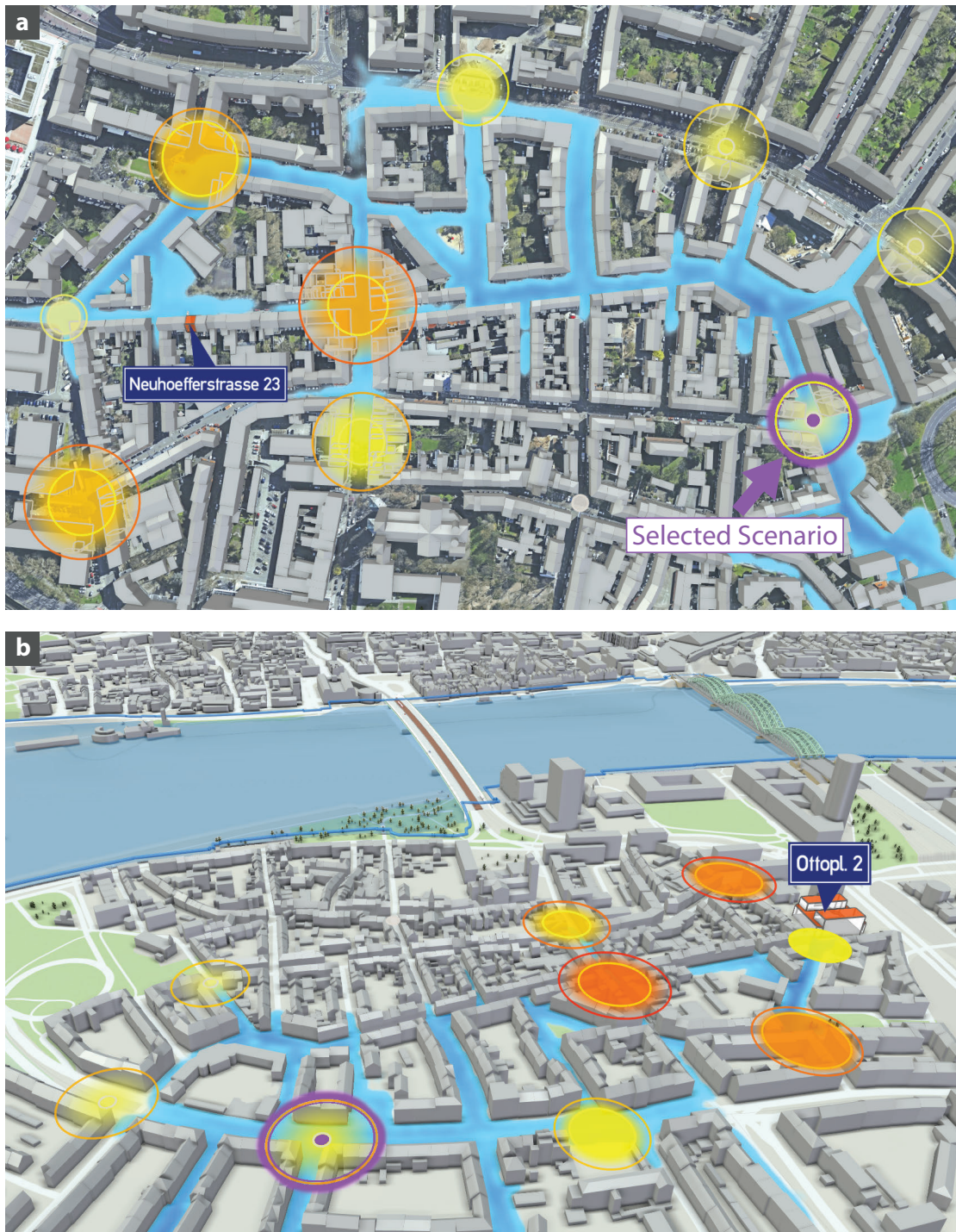
Figure 2.7: Sewer overflow scenario selection. (a) One sewer location is selected, the shades of blue show the expected water depths associated with an overflow at that location. (b) A different perspective is chosen, and a different building is selected.

2.8   Floodwall Breaches

The object-centered vulnerability to floodwall breach incidents is presented with a plot along the actual floodwall in the 3D city model, as shown in Figure 2.8. For each position on the floodwall, the plot displays the vulnerability to a breach occurring at that particular position. The visualization uses the pattern described in Section 2.5 varying along the floodwall. Namely, for expert users, the representation includes the minimum and maximum within the uncertainty range as two lines, and the expected value is encoded in the color of the area. Furthermore, the uncertainty of the expected value is conveyed with a transparency gradient. For non-experts, the plot shows (with both color and magnitude) only minimal, average, or maximal vulnerability across the relevant scenarios. As for the sewer-overflow vulnerability visualization, the context legend is displayed on mouse-over.

To convey further vulnerability details, the possible breach widths for every considered breach position are visualized on the other side of the floodwall by means of a centered bar plot. The extent of each bar corresponds to the width of a breach the bar represents. The shapes of the bars follow the outline of the floodwall at the corresponding positions. The color of each bar encodes the average vulnerability across all scenarios sharing that particular breach position and width. In other words, such aggregations are done over the two remaining ensemble dimensions, i.e., water levels and durations. The user can click on any bar to explore the expected inundation, aggregated over the relevant breach scenarios.

Figure 2.8a illustrates the design described above. For the building of interest, the dangerous breach positions are clearly visible from the vulnerability plot. For another building selected (see Figure 2.8b), the vulnerability profile changes. Figure 2.8c shows the close-up of the same case with the context legend displayed, which shows the 0 % and 100 % vulnerability bounds. At some positions, a breach of any size is very dangerous, but there are also positions where only some or none of the breach sizes are dangerous for the building of interest. In Figure 2.9a, the vulnerability of the same building with respect to the accessibility of the hospital is visualized. A scenario is picked (purple), for which the aggregated water depths map is shown. Figure 2.9b shows the non-expert visualization for the case of Figure 2.8a. The worst-case vulnerability is displayed with respect to the building damage. There is a larger set of potentially dangerous breach positions (right-hand side of the view). For the picked scenario, the worst-case water depth map is shown.

2.9   Accessibility of Important Infrastructure

In this section, we describe our visualization of the impact of flood-related hazards on the accessibility of important infrastructure with respect to the building of interest. By important infrastructure we mean buildings such as hospitals, pharmacies, or schools (see Figure 2.10), as well as the routes by which these buildings can be reached from
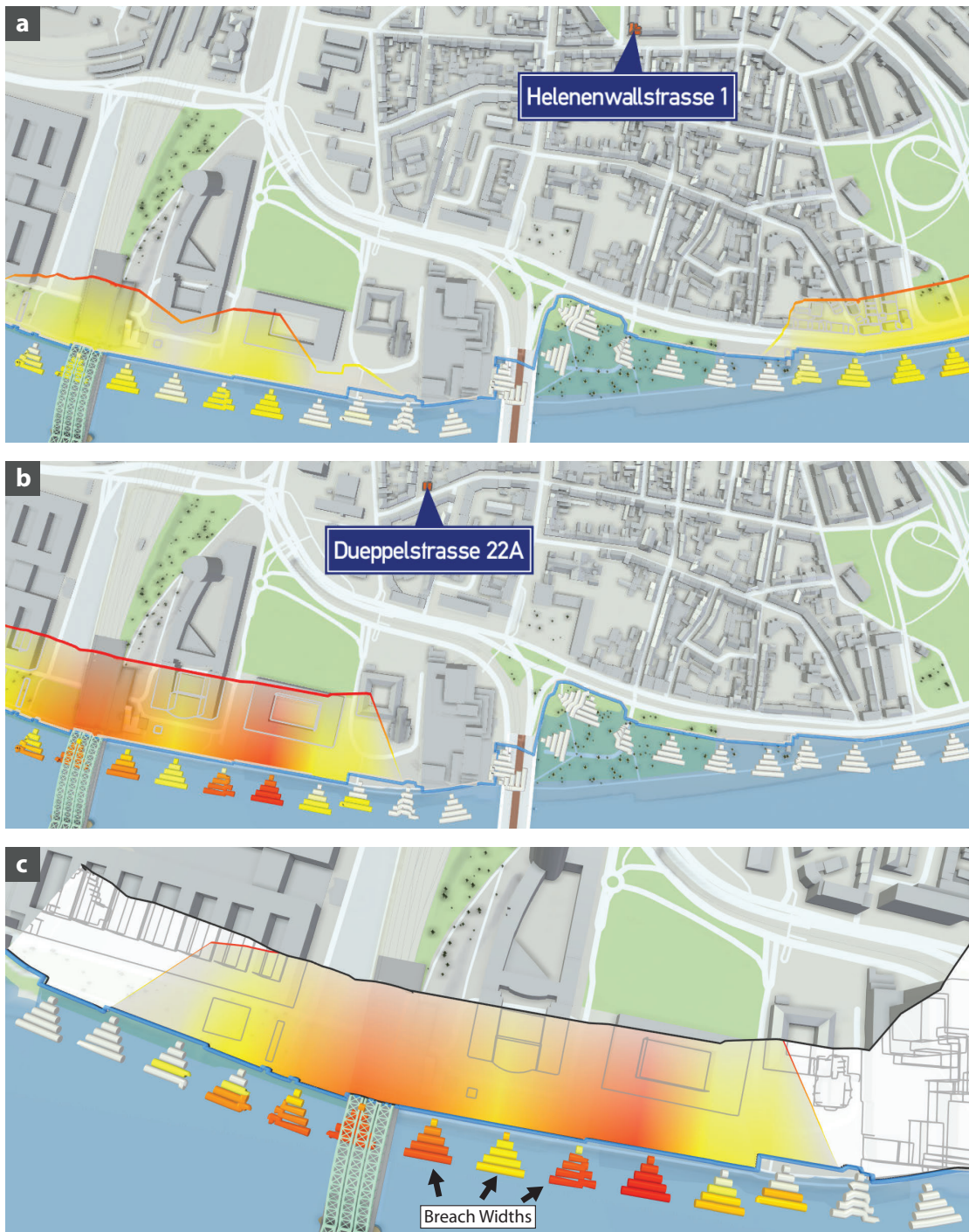
Figure 2.8: Vulnerability to floodwall breaches. (a) For the selected building, dangerous breach positions are indicated by the plot. (b) Another building has a different vulnerability profile. (c) A close-up view of (b) with the context legend displayed. The vulnerability per width is shown for each position in centered bar plots.

Figure 2.9: Floodwall breach scenario selection. (a) Vulnerability with respect to the hospital accessibility. A scenario is selected by clicking on a width bar, showing the aggregated water depths on the terrain. (b) Non-expert view of vulnerability with respect to the damage of the building of interest. The worst-case values are shown in the plot and width bars. The worst-case water depths are displayed for the selected scenario.

the selected building (see Figure 2.11). Since there are usually multiple ways to reach a location, we evaluate ten distinct routes to each of the important buildings. We automatically request these routes from the Google Directions service. To create a diversity, we make ten routing requests through ten different way points for each important building. It is likely that, within the whole set of routes obtained, many route parts overlap. Since our visualization does not support overlapping routes, we split the routes into segments to isolate the overlapping parts and then remove the duplicate segments.
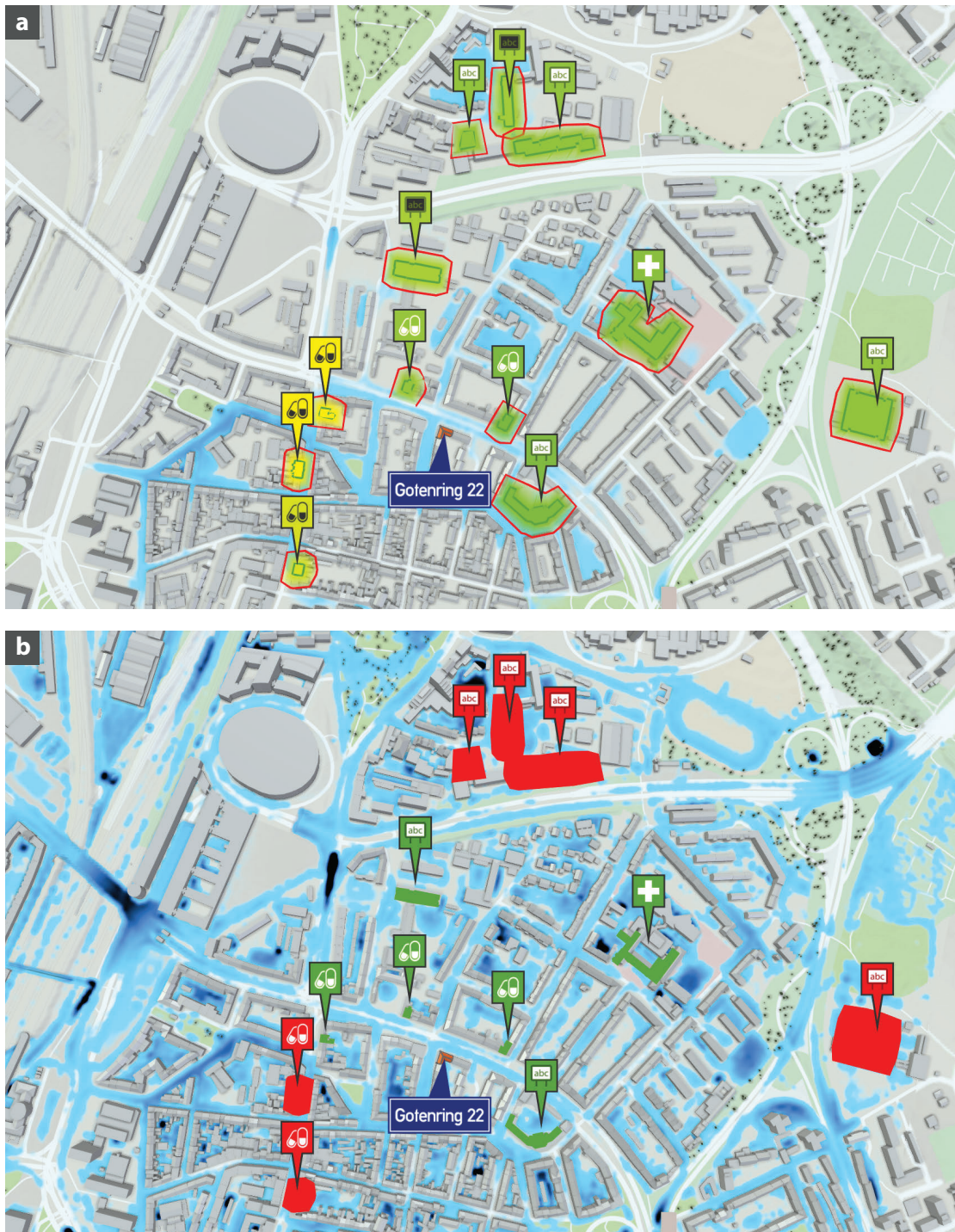
Figure 2.10: Building accessibility in two different scenarios. (a) Expert view showing the accessibility of hospitals, schools, and pharmacies with respect to sewer overflows. (b) Non-expert view showing the worst-case accessibility with respect to heavy rains.
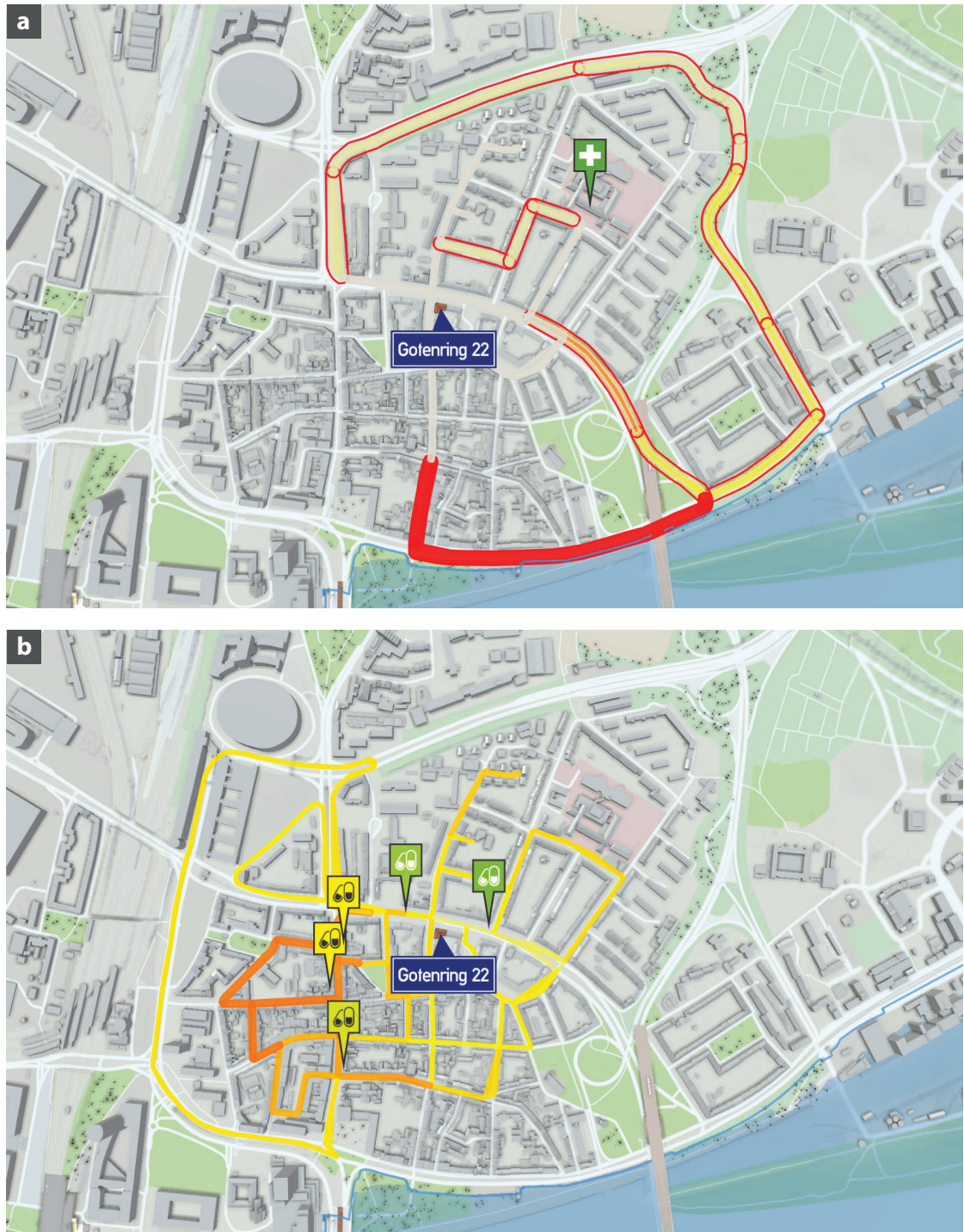
Figure 2.11: Route accessibility with respect to the shown buildings. (a) Expert view showing the hospital accessibility with respect to heavy rain incidents. (b) Non-expert view showing the expected accessibility of pharmacies with respect to sewer overflow scenarios.

As a result, we obtain a set of unique route segments from which any of the routes can be reconstructed. Given a scenario, we consider a building accessible if there exists a connected sequence of route segments leading to this building such that each segment in this sequence is accessible. A segment is considered accessible if the water depth along it does not exceed 0.3 m.

The accessibility is displayed by using the pattern described in Section 2.5. Figure 2.10a shows the impact of sewer overflows on the accessibility of multiple important buildings (marked with labels) with respect to the building of interest (marked with an address plate). In this visualization for expert users, the 0 % vulnerability bound is given by the building contour, whereas the 100 % bound is shown on mouse-over. One can see that each building is accessible in most scenarios, yet inaccessible in some worst-case scenarios. In addition to the impact, the average water depths across all relevant scenarios are shown. In Figure 2.10b, the impact of heavy rains on the accessibility of the same buildings is shown along with the maximum water levels across the relevant scenarios. Here, the worst-case accessibility and the worst-case water depths are displayed for non-expert users. Apparently, the green buildings are always accessible, whereas the red buildings are inaccessible in some scenarios.

In Figure 2.11a, a visualization for expert users is presented, showing the impact of heavy rains on the accessibility of routes to the hospital. In this case, the hospital is unreachable over the red routes in every scenario, certainly reachable over the gray routes, and mostly reachable over other routes in the vicinity. Figure 2.11b shows the average impact of sewer overflows on the accessibility of routes to pharmacies in the vicinity. The presented visualization targets non-expert users. Route segments display solid colors according to their accessibility. The average accessibility of the buildings is indicated by the label colors only. The full accessibility visualization on top of the buildings is omitted to avoid visual clutter when showing the routes.

2.10 Evaluation

The overall idea of the presented work developed from a long-lasting and well-established collaboration with the flood protection center of Cologne, Germany. Two experts, a flood manager and a logistics expert, were then evaluating our solution in two separate sessions. Both experts are also partially involved in public communication activities. After the introduction to the basic concepts, the experts were asked to interpret different visualization results and evaluate their usefulness and readability. During this evaluation, valuable feedback was provided, and suggestions were made on how to refine our solution. We used these suggestions to improve our solution after each session.

Both experts required a learning phase of about 15 minutes, and multiple different examples for comparison, to correctly interpret our uncertainty visualization pattern. The façade area plot and interactive 2D chart were found to be more comprehensible than mapping uncertainty onto 3D visualizations. Initially, both experts assumed that the size of the mapped vulnerability visualizations (for example for sewer overflows) had

a spatial relation, and interpreted them as areas of influence. After the main idea of these visualizations was internalized, the domain experts were able to correctly interpret the visualizations and found them useful for their needs in flood management. However, they stated that the proposed uncertainty-aware visualizations contained information unnecessary for the general public and were hard to interpret without knowledge of statistics. According to our experts, for the general public, a binary information on whether or not the object of interest is vulnerable would have been sufficient. Therefore, they suggested to simplify the visualizations, which led us to implement two different user models, i.e., one for flood managers and one for the general public.

The façade area plots were found helpful and comprehensible for both targeted user groups. The experts agreed that the mapping of water levels onto the building façades is very descriptive and immediately tells the general public how vulnerable the object of interest is. It was also pointed out by the flood manager that this allows affected people to focus protection measures on façades where the highest damage is expected. Estimation and visualization of cellar flooding was also well-received by both experts. According to them, the combination with interior reference objects gives a vivid image of the possible damage. The exterior reference objects were rather perceived as decoration that is too far away from the actual building to be helpful. This led us to displaying a door on the façade. The façade line plot was seen as useful for experts, but too complex for the general public. Both experts were very positive about the cutaway visualizations of terrain and surrounding buildings, since it helped to avoid occlusion of the building in focus. Aligning the address plate to the façade of the building rather than using a floating label had no impact on the perception.

Generally, the visualizations including our uncertainty pattern were considered useful. The interactive 2D chart was seen as intuitive and well-suited for exploring the scenario pool even for the general public. The spatially mapped vulnerability visualizations were found useful mostly for domain experts. The logistics expert welcomed that the shape of the visualization resembles the shape of the object it is mapped onto. He also stated that displaying the contours of the underlying buildings on top of the vulnerability plot helps with the perception of the locality. The other expert, however, found the contours distracting. It was well-received by both experts that the breach- and sewer-related visualizations allow flood managers to focus on particularly dangerous regions. The display of breach widths was seen as a helpful addition. The interactive scenario picking from these visualizations was found intuitive. Both domain experts appreciated the indication of the picked scenario and the display of the corresponding water depths map. These were found comprehensible and aesthetic. Visualizations mapping accessibility onto streets and important objects were highly rated. According to the experts, this is an intuitive way to convey the accessibility of important infrastructure to the general public.

To summarize, the domain experts considered our solution useful both for their tasks in flood management and for the general public. The visualization of uncertainty was found reasonable for domain experts, but challenging for non-technical users. The experts concurred that, for the general public, simplified visualizations were needed to conceal the

uncertainty, and suggested to consider two different user models. This major suggestion was implemented and made our application more versatile.

2.11 Conclusions and Future Work

In this chapter, we addressed the problem of isolating geospatial information for a particular object scattered in GIS data and a large pool of pre-simulated flooding scenarios. The automatic combination of these heterogeneous data in a conclusive way is hidden from the user and relieves him or her from dealing with multiple and often incomplete data sources. As a consequence, users of our application require no knowledge about the underlying data and data sources. The object-centered approach enables the user to obtain expressive visualizations of flood-related vulnerabilities and impacts with respect to a particular building. This allows both the general public and domain experts to investigate, whether, why, and how much this building is in danger, without manually exploring different flooding scenarios. Although our solution was tailored to the needs of flood management, the proposed concepts and visualizations also work for any other hazard to infrastructure, for example wildfires or landslides.

For the visualization of vulnerabilities and impacts, we rely on visual metaphors lay people are used to, like city maps or temperature ensemble figures in weather forecasts, and extend them. Whenever possible, information is mapped onto the spatial domain to establish visible relations between geospatial and abstract information. Through different user models, the uncertainty can be visualized in a user-specific manner to satisfy the different demands of experts and non-experts. Yet a meaningful, compact visualization of uncertainty remains a challenging task in the visualization for the masses.

The evaluation has also shown that mapping abstract information to the spatial domain can be misinterpreted. This happened in particular with the vulnerability to sewer overflows and floodwall breaches, where the area covered by the respective visualizations has been interpreted as the extents of flooding. To establish a correspondence between the abstract vulnerability of the building of interest to flooding incidents, we anchor the visualizations at the location of these incidents in the spatial domain. However, this is inconsistent with the impact visualizations where the mapping creates a direct correspondence between spatial locations and adverse impacts. Therefore, a more comprehensible visualization might be achieved by connecting the locations of incidents and the building of interest visually with paths or arrows [BSV11, Leh18] on which the uncertain vulnerabilities can be mapped, e.g., by color and thickness, but this requires further evaluation.

A next step for future work is to give users the ability to adapt to the vulnerability by means of interactive sketching of protection measures such as barriers, local terrain modification, and curb elevation. The coverage of the adaption can then be visualized on top of the vulnerability to see which incidents can be addressed with the protective measures. Since such protective measures require a re-simulation of the scenario pool, strategies for an efficient re-computation are needed, such as considering only dangerous

scenarios. Another field of future work is the integration of more complex ensembles to better represent the real world, for example combinatorial ensembles of sewer locations and heavy rain. A further example would be the visualization of cascading events such as heavy rain, which eventually leads to sewer overflows. Apart from the technical challenge, there is no general concept on how to combine multiple ensemble dimensions for the extraction of the relevant information in a comprehensible visualization. It is possible in some cases, as done with the breach positions and widths for the visualization of vulnerabilities to floodwall breaches, but in general, this is an open question. Right now, our solution only focuses on direct impacts of flood-related hazards. For a more thorough investigation, long-term impacts of erosion and frequent flooding incidents to the infrastructure should be considered as well. In this context, financial considerations could also play a role when estimating the cost for maintenance and repair of said infrastructure compared to the cost of adaption. To know whether proper protective measures are worth their investment concerns both domain experts and the general public.

3

Composite Flow Maps

3.1 Introduction

Flow maps have long been used to illustrate the movement of objects between locations. One prominent application of flow maps is the migration of population between multiple geographic regions. Many fields where users have to explore massive movement data can benefit from using flow map techniques. For logistics, delivery flows between sources and destinations are of interest. Modern architectural planning requires consideration of possible evacuation scenarios, where flow maps can represent people heading for emergency exits. In flood management, the flows of heavy rain or flood water are important as they may carry potentially dangerous debris. In all of these cases, for effective decision support, interactive exploration of multiple flow components at once is needed. The flow components may be given by two opposite transportation directions, alternative scenarios in consideration, different kinds of materials being transported, or people evacuating from specific rooms. At different stages of scenario assessment, the decision maker requires either an overview of general flow trends or a detailed representation of local features. We suggest that such levels of detail should be driven by the application semantics. While uninteresting local flows may be merged to reduce visual complexity, important details should still be preserved and possibly highlighted using additional overlay visualization.

When visualizing material transport, flow maps offer several important advantages over pathline or streamline techniques, such as the ability to present quantitative and directional information while keeping the visual space free of occlusions. For discrete entities such as delivery trucks, it is relatively easy to keep track of the transported quantities. However, in case of the movement of a continuous medium such as water, the computation of material flows requires numerical integration over a given finite surface. Therefore, a generic flow map computation algorithm should be based on zones connected to each other. In this case, the material is abstracted to move between the discrete zones rather than in a continuum. This approach is applicable for both discrete and continuous
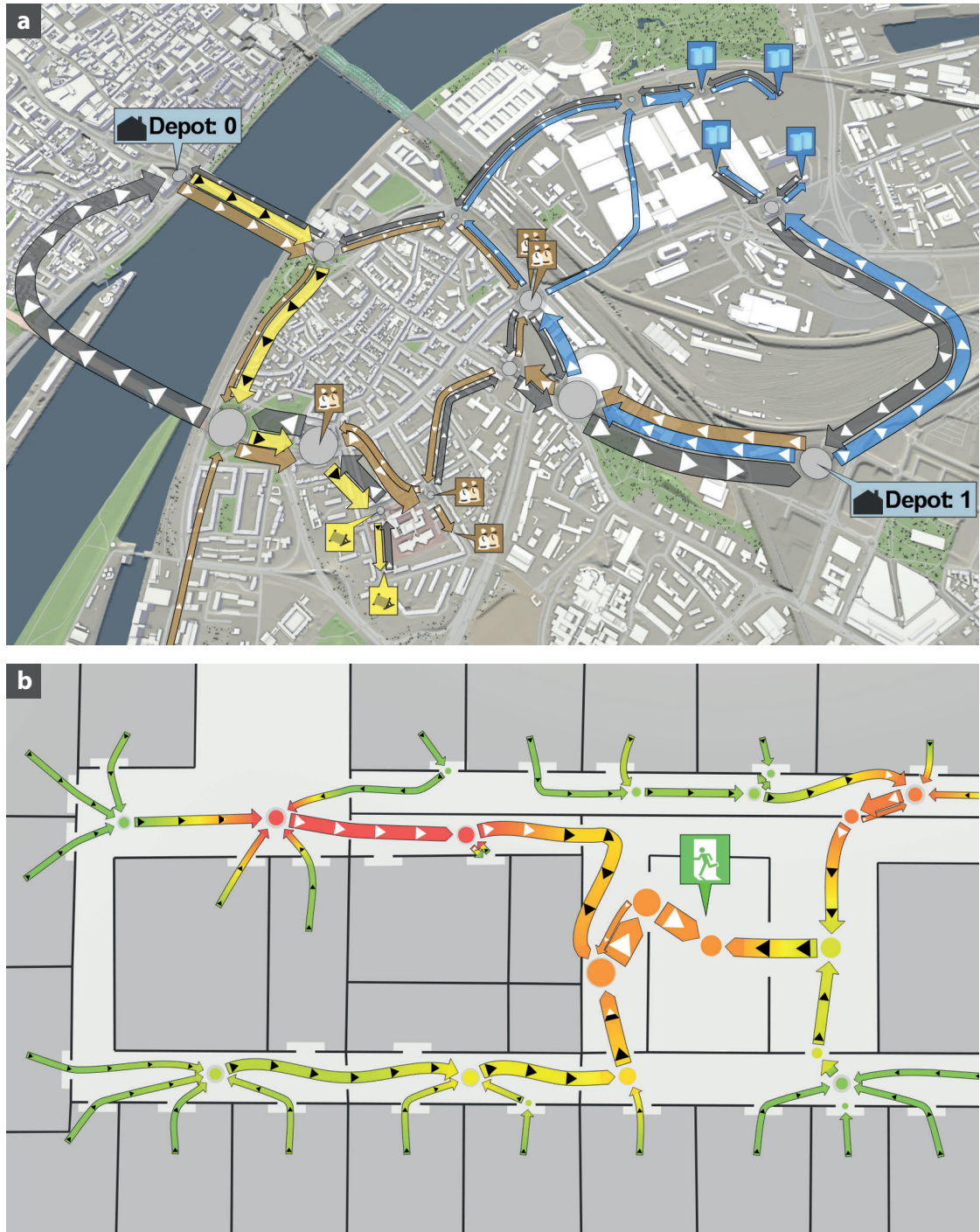
Figure 3.1: Quantitative visualization of material movement for several flow components at once. (a) Different materials in a logistics delivery process. (b) Flow map for crowd movement in an evacuation scenario, colored by the time of stay.

materials as it considers the transport through zone-to-zone interfaces and thus naturally provides means for flow computation. Moreover, splitting the space into zones breaks down the potentially complex flow data into easier to handle flows between zones while keeping the important reference to the geographic space.

In this chapter, we propose a technique for the automatic generation of flow maps from large movement data obtained by simulation. Multiple flow components are combined in one visualization by means of ribbons representing different materials, directions, or flows related to particular origins or destinations. Alternatively, such composite flow maps can display information from different scenarios. Our technique is based on splitting the domain into multiple zones, where the zones are derived from application semantics and the geospatial context. After this zonation, we compute the flows between adjacent zones. From these, the flow map is generated. The presented technique enables the visualization of composite flows between an arbitrary number of zones in both directions. Additionally, our flow maps support varying levels of detail driven by the geospatial semantics inherent in the application. Irrelevant local features can be generalized, whereas important details are preserved and possibly highlighted.

We demonstrate and evaluate our technique in real-world applications. The first application is concerned with transportation logistics for constructing flood protection barriers (see Figure 3.1a). To plan the routes for reliable material deliveries, an overview of multiple time-dependent processes is needed. The second application deals with the planning of sewer inlets. The sewers must be able to consume water without overflows even under extreme weather events. To effectively place new inlets, the expert needs an understanding of the local flow behavior, including the water exchange of the surface with the sewer network. In the third application, the evacuation from an office space is modeled (see Figure 3.1b), where multiple rooms are connected to an emergency exit via shared corridors. The placement of doors and planning of room occupancies require considering possible evacuation activities. People should be able to quickly leave the area without getting stuck or jammed.

In summary, the scientific contributions presented in this chapter are:

- A composite flow map layout algorithm where components are different directions, materials, scenarios, origins, or destinations

- Context-aware and importance-driven levels of detail

- Highlighting of problematic areas with overlay visualization of spatial data crucial to the application

- Semantic zonation for flow computation
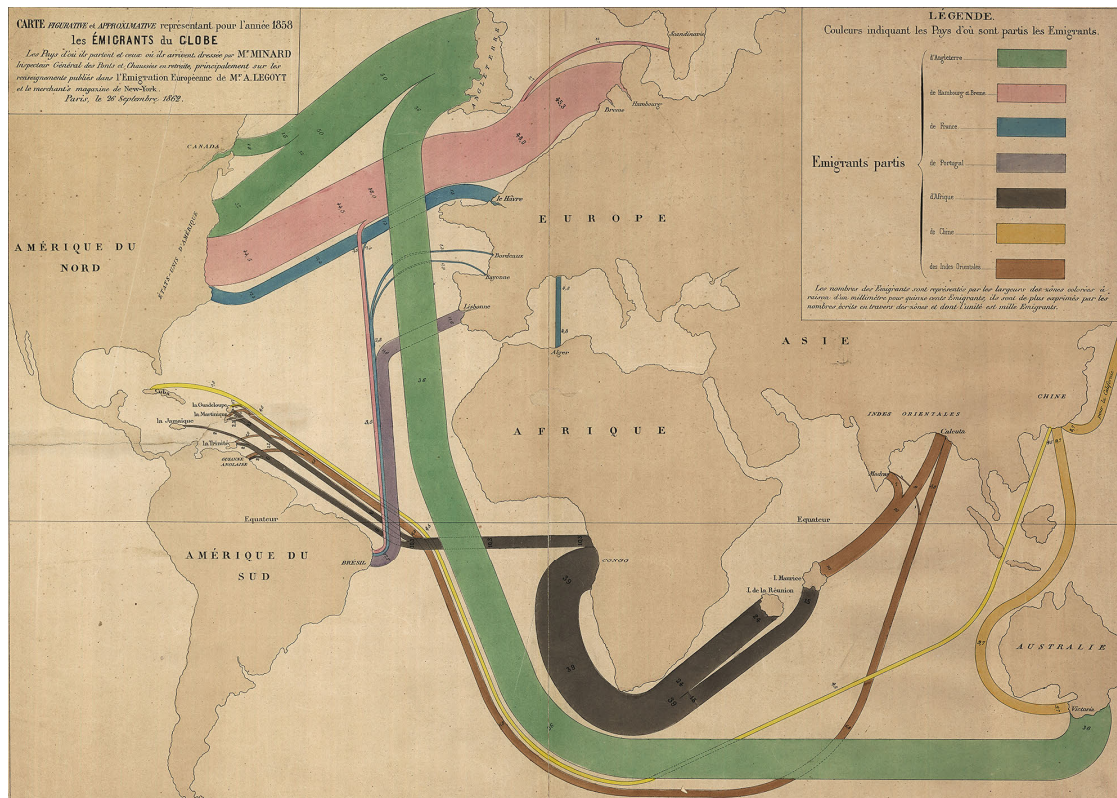
Figure 3.2: Hand-drawn map of emigration by Charles Minard.

Copyright: Charles Joseph Minard, public domain, commons.wikimedia.org/wiki/File: Charles_Joseph_Minard_-_%C3%89migrants_du_Globe.jpg, last visited on January, 18th 2020.

3.2   Related Work

A *flow diagram* is a visual representation of the flow of a quantity or the succession of events in a system [Har96]. It consists of nodes representing the elements of the system connected by arrows representing the flow with quantities mapped to widths. For example, the nodes could be cities, while the flow could represent people traveling from one city to another. Further examples include the flow of data through a processing pipeline, and the transfer of energy between components of a mechanical system.

Flow maps are flow diagrams frequently encountered in the context of cartography, or, in general, whenever the nodes represent geographical locations. For instance, Figure 3.2 shows a hand-drawn flow map depicting emigration patterns. Colors denote the country of origin, while the width of the arrows gives an indication of the magnitude of emigration. Notice that the illustrator carefully chose colors, layouts, and compositing rules in order to depict multiple flows simultaneously.

Due to their effectiveness in depicting quantitative information, flow diagrams have been investigated thoroughly in the context of visualization. The main challenge in the creation of flow diagrams is to determine an appropriate layout of nodes and arrows. The flow between two locations can be easily depicted by a straight arrow between them [Tob87]. But, as the number of location grows, occlusions between arrows increase. Phan et al. [PXY+05] address this issue by applying hierarchical clustering to the set of geographical locations. Arrows directed towards the same cluster are bundled together, and they can be bent in order to avoid occluding other clusters. Buchin et al. [BSV11, BSV15] define a set of quality requirements for flow map layouts, such as avoiding edge crossing and minimizing the length of arrow segments. In their approach, the diagram can include multiple tree structures, each one showing the flow from one source to several destinations. The layout is determined by optimizing a number of quality measures. The work by Andrienko and Andrienko [AA11] instead exploits an automatic tessellation of the 2D domain. They extract and cluster characteristic points from the input trajectories. The centroids of the clusters define a Voronoi diagram. Flows are computed by counting trajectory segments between adjacent Voronoi cells, separating opposite flows. The result is then displayed using straight arrows. Since the underlying graph is given by connectivity of Voronoi cells, arrows are guaranteed to never overlap. This work is mainly focused on depicting motion over geographical maps. Andrienko et al. [AABW12] analyze the effectiveness of analogous flow map visualizations for studying eye gaze movements. Multiple flows within the same flow map are taken into account, but no dedicated strategy is adopted to avoid overlaps between them. A detailed survey on the visual analysis of movement data can be found in the book by Andrienko et al. [AAB+13]. Guo [Guo09] combines flow maps with other techniques to visualize multivariate data.

The data structure underlying a flow diagram is, in most cases, a weighted graph. An abstract visualization of a dense graph in general can be produced by *edge bundling*. Holten [Hol06] proposes a technique for bundling adjacency edges within a hierarchy. He adopts the path between two nodes of the hierarchy as the control polygon of a spline, therefore obtaining smooth bundles of curves. Since a hierarchy is required, this method cannot be applied to general graphs. This is addressed by Cui et al. [CZQ+08], who facilitate a regular grid defined on the 2D domain. Edges with similar orientation are grouped together within the grid cells. Voronoi diagrams and quadtrees can be combined to generate an adaptive grid for routing edges [LBA10, EHP+11].

Iterative procedures can be employed as well. Hurter et al. [HET12, HET13] treat the 2D domain as an image. A density-based representation of the graph [DV10, SWV+11] is produced via Kernel Density Estimation [Sil86]. A number of image processing filters are applied in order to detect ridges and local maxima in the image. Then, edges are iteratively shifted towards these topological structures, therefore obtaining the desired bundling. This approach is further extended by Peysakhovich et al. [PHT15]. A physics-based approach is proposed by Holten and van Wijk [HVW09]. For each pair of edges, a similarity measure is computed. Dummy nodes are inserted along edges. If two edges are sufficiently similar, dummy edges are inserted between their dummy nodes. The desired

bundling is obtained by running a force-based simulation that treats dummy edges as springs while also minimizing the variation in curvature.

Edge bundling cannot be directly used to create flow maps, since quantities are neither accurately represented nor conserved. This is addressed by Debiasi et al. [DSD14] with a force-based technique for generating flow maps from one root node to several leaf nodes. This technique enables the supervision of the flow map layout during the creation process. Flows can be aggregated, straight edges can be reserved for the main flow channels, and overlaps of edges with other nodes and edges can be avoided. For these reasons, we adopt the work by Debiasi et al. as a starting point for our flow map layout algorithm.

3.3   Flow Map Generation Pipeline

In this section, we give a brief overview of the flow map generation pipeline. The table in Figure 3.3 summarizes the steps. As a prerequisite, we assume that the input datasets are given. These are the actual material movement data as well as the corresponding geospatial datasets, available from public GIS services or the competent authorities.

In the first step (see Figure 3.3a), we derive context-aware zones within the domain of interest from the input datasets and the semantic information contained in them. Depending on the application, zones can be one- or two-dimensional (see Figure 3.3a, left and right columns, respectively). We then compute *border flows* between all adjacent zones (see Figure 3.3b). Border flows are the aggregated material quantities transported from one zone to another one over the common interface in the considered time range. Two zones are considered adjacent if their geometric representations share an endpoint in the one-dimensional case or a polygonal chain in the two-dimensional case. As a result, zones and their connectivity are represented by an abstract zone graph with vertices corresponding to zones and edges carrying the zone adjacency information (see Figure 3.3c). The border flows are attributed to these edges and comprise the components for each relevant material in both directions separately. Finally, the spatial embedding of the zone graph is computed (see Figure 3.3d), where the zone graph vertices are mapped to spatial locations. The graph is then simplified according to the context-based level of detail by iteratively merging the vertices and aggregating the corresponding flow values. The flow component ribbons are generated according to the border flow values and are spatially arranged to avoid overlaps. Additional overlay visualization may be added to highlight important information on top of the ribbons.

For a concrete example of the flow map generation, the logistics application is considered (see Figure 3.3, left column). First, a zone representation of the street network is created by splitting up the street lines at all intersections (a1). Border flows between connected street zones are then determined by counting the number of trucks moving from one street to the next one (b1). The movement data are time-dependent, so movements between streets are accumulated over all time steps up to the time step of interest. From the zones and border flows, a directed acyclic graph is derived (c1). Zones are represented as vertices and are connected to adjacent street zones with weighted edges, where the
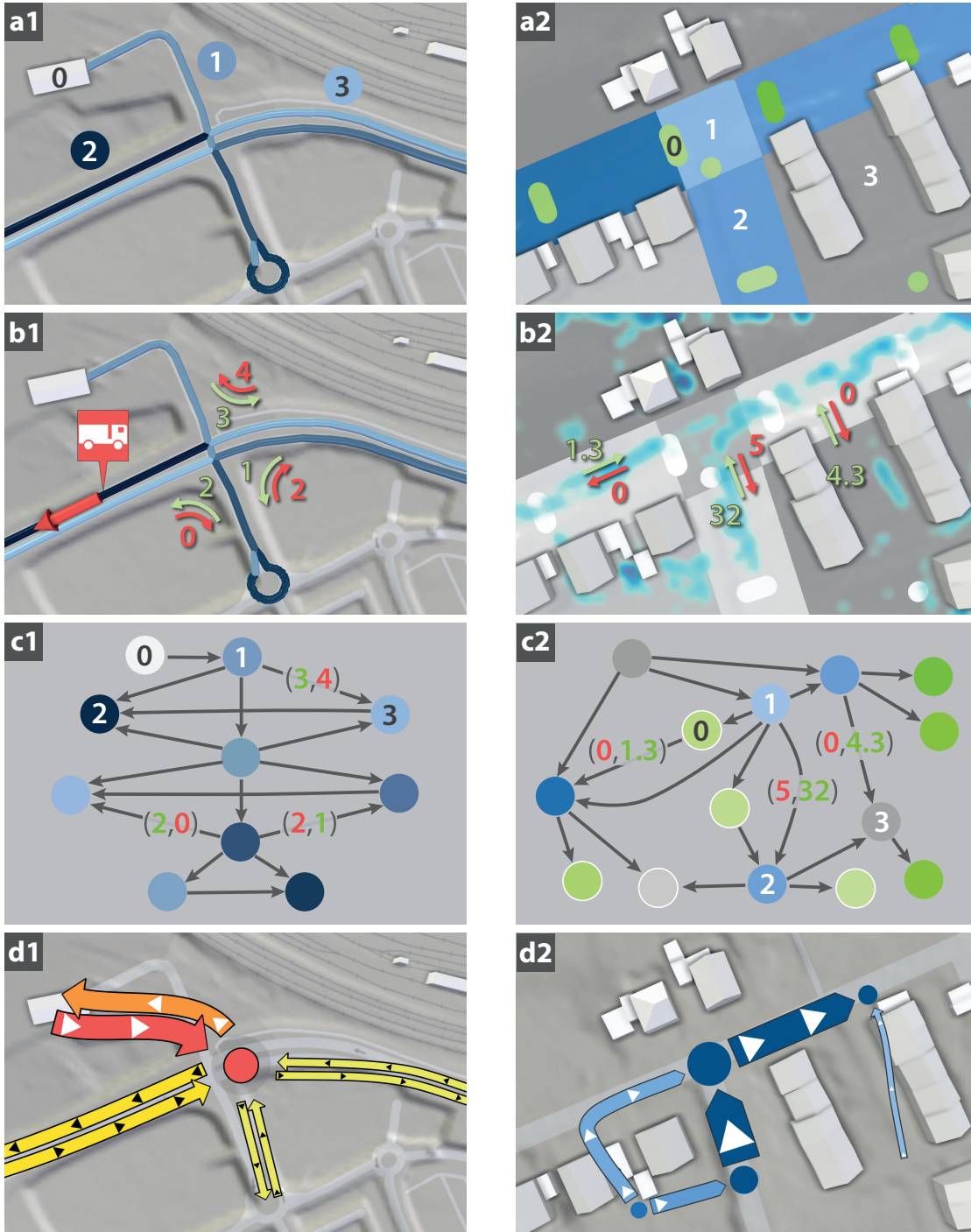
Figure 3.3: Flow map generation pipeline for 1D (left column) and 2D (right column) flow data. (a) Semantic-based zonation. (b) Bidirectional border flows between example zones (red, green) are derived from simulation data. (c) Abstract, directed acyclic zone graph. (d) Resulting flow map visualization.

weights are the border flows. For example, a border flow of (3, 4) between vertices 1 and 3 means that three trucks moved from Street 1 to Street 3, while four trucks moved from Street 3 to Street 1. The abstract zone graph (c1) serves as the common representation of the flow in all applications, from which the geospatial visualization (d1) is derived.

3.4   Data Acquisition

In this section, we give the details on the required input datasets and briefly describe the simulation models used to produce the original movement data for the three considered applications. The water and logistics applications consider real-world scenarios in the urban area of Cologne, Germany. Both require a digital terrain model, building shapes, and the street network for providing a geospatial context and deriving meaningful zones. While the high resolution terrain raster data necessary for flood simulation need to be requested from the competent authorities, building shape data are openly available through a public GIS service [Off]. For each application, additional specific data are required, which are described in the following. For the logistics application, delivery routes and travel times are automatically obtained from the Google Directions service, and the traffic data is available through the Open Data Cologne service [Off]. In the water application, to model the sewer system, we use the shape data describing the locations of existing sewer gullies. These data were provided by our collaboration partners from the flood protection center of Cologne. The evacuation application is based on a manually created arrangement of rooms and corridors resembling the authors' office space. Importing an existing domain layout in the Building Information Modeling format (BIM), for example, can be implemented with little effort.

Based on the acquired datasets and on the specified initial and boundary conditions, such as demand and supply for logistics, inflow hydrographs for water, or initial agents' placement for evacuation, we simulate the actual movement data to be represented with our composite flow maps. We now give a brief overview of the simulation procedures.

Logistics Simulation   For the logistics application, the flow map is meant to represent the amount of different materials transported by trucks between different locations. Given a set of depot locations and a number of trucks available, a logistics scenario is defined by delivery locations and the demand for each material type at each of them. To compute a delivery plan, we use a greedy heuristic [WKS+14]. In short, any trip of any truck between a depot and a delivery location or between two delivery locations is considered a single job. The duration of each job is calculated from the loading and unloading times and the actual travel time. The loading and unloading times depend on the material amount and types. Travel times and routes are retrieved from the Google Directions service. The greedy heuristic assigns jobs to trucks until the demand at all locations is fulfilled. From all jobs available to a truck, the one that is expected to finish first is assigned. This heuristic provides no optimal, but practicable delivery plans. The output of our logistics model is a list of jobs per truck where each job constitutes of a route,

delivery durations, and transported material amounts. From these results, georeferenced flow components for individual materials can be extracted and visualized with a flow map.

Water Simulation   For water movement, the key quantity for deriving the flow map is the so-called hydrological discharge, which is the amount of water transported through a given cross-sectional area over time. For simulating the surface run-off, we use a GPU-based implementation of a two-dimensional numerical scheme solving the shallow-water equations [HWP$^+$15]. The emission of water is done either by specifying a hydrograph at a simulation boundary (for a breach) or by evenly adding fixed, predefined amounts of water to all simulation cells at every time step (for precipitation). The sewer system is modeled with a capacity-based heuristic where, instead of actually simulating water flow through sewer pipes, only hydrostatic forces are considered. The overall capacity of the sewer system constitutes of the proper capacity modeled as a large tank and the capacities of the gullies. Initially, the tank is filled with a certain amount of water, and constant inflow and outflow are given. These are intended to model the routine sewer system usage of the city. The water consumed from the surface by each gully contributes to filling the tank. In case of precipitation, the water emitted at the rooftops is also assumed to go directly into the tank. When the tank is filled, the whole system acts as a set of communicating vessels, where the vessels are given by individual gullies. Until a gully is filled, it can only consume water from the surface. However, if a gully is full, it can emit water as well. The amount of water emitted or consumed by each sewer gully is defined by the relative elevation of each gully in the sewer network and by the correlation of the water depths above all gullies that are full, i.e., from the pressure of the corresponding water columns. The coupling procedure for the surface flow simulation and sewer modeling ensures that the simulation grid cells at the gully locations exchange the water with the sewer system in a consistent way.

Evacuation Simulation   Assessment of evacuation scenarios requires knowing the numbers of people taking different ways as well as locations where they eventually gather. To get this information, we use an agent-based pedestrian simulation driven by a force-based heuristic. The agents are initially distributed in the rooms and assigned a target location that can only be reached by passing through emergency exits. In the beginning, the agents compute their optimal ways to the target location. Subsequently, at each time step, every agent evaluates the movement of other agents in proximity and reconsiders its planned way towards the target.

3.5   Data Preparation

This section gives a more detailed description of the steps *a–c* of Figure 3.3. These steps are dedicated to data preprocessing before the actual flow map creation. Step *d* is thoroughly described in the next section.

3.5.1   Semantic-Based Zonation

With all input datasets available, the automatic zonation is performed (see Figure 3.3a). Multiple connected zones are derived in the domain in order to compute border flows between them. In contrast to other related efforts [AA11], we do not use a tessellation of the domain based on the analysis of the material movement data, but rather derive meaningful zones from the geospatial context. We argue that arbitrary zonation might hide important information from the user while exposing unimportant details. For example, for planning sewer gullies, it is crucial to understand how much water is consumed or emitted by each gully, whereas water flows between different parts of the same street are irrelevant. For indoor evacuation planning, the important transitions are made from rooms through doors to corridors, with doors being potential bottlenecks. These two examples suggest to tailor the zonation to particular application cases, for example to distinguish street zones from sewer zones for the first example, and room zones from door zones and corridor zones for the second one.

Depending on the application, the required zonation can be in one or two dimensions. In case of overland logistic deliveries, it is reasonable to assume that materials or goods are transported through streets only. Therefore, it is sufficient to decompose the street network into one-dimensional zones, where each zone is a line representing a street (see Figure 3.3a1). Implementation-wise, such zonation can be obtained from a set of lines describing the street network in the domain of interest or, as an optimization, only the subset corresponding to the streets utilized for actual deliveries. The lines are split at their intersection points and duplicates are removed. In the resulting set of unique, connected line segments, each element constitutes a one-dimensional zone (see Figure 3.3a1).

Crowd or water can move freely across the surface. In such cases, two-dimensional zones are required that fully cover the domain (see Figure 3.3a2), taking into account shape data describing the domain. For large-scale urban scenarios, these can be, for example, street shape lines, land use polygons, or sewer locations. In recent years, such geospatial data have become increasingly comprehensive and freely available online. In the water-related example, three types of zones are needed (see Figure 3.3a2). These are street zones (blue), sewer zones (green) and block zones (grey). The zonation algorithm starts by splitting all street lines at their intersection points and removing duplicates. For each of the resulting unique, connected street line segments, a spatial buffer with an empirically determined radius of 9.9 m is created around it. Similarly, circular buffers with a radius of 2 m are created for each point describing a sewer gully location. The outlines of all buffers are split at their intersection points. Each zone in the domain is marked out by a unique set of connected segments of spatial buffer lines. Identifying these can be reduced to the problem of finding all faces of a planar graph. We interpret each endpoint of each segment as a graph vertex. For any two vertices of the graph, there is an edge connecting them if and only if there exists a line segment with the two corresponding endpoints. Since all segment intersections are already eliminated, the resulting graph is guaranteed to be planar. The planar faces of this graph exactly correspond to the desired zones. For

the planar faces computation, The Boost Graph Library [Boo] is used. From the faces, the actual geospatial zones can be easily reconstructed and labeled as street zones (inside street line buffers), sewer zones (inside the sewer buffers) and block zones (the other zones). For indoor applications such as evacuation, wall lines and door locations are needed, as well as semantic annotations for rooms and other areas. The actual zonation process is similar to the one described above.

3.5.2 Zone Graph and Border Flows

The actual flow map is generated from border flows, which are computed using the zonation and the material movement data. Border flows are the material quantities transported between each two adjacent zones through their common interface in the considered time frame. Since the transported materials can be either continuous or discrete, we distinguish two cases. In case of a continuous medium (for example water), the quantity is represented by the volumetric flow rate of the transported medium. For water movement, this value is equivalent to the hydrological discharge between the zones (see Figure 3.3b2), which we compute numerically. At each time step, the value is multiplied with the time step size to calculate the actual material volume transported in that time step. For each zone-to-zone interface, such quantities are accumulated over all time steps to obtain the overall material quantities. In case of movement of discrete entities (for example people or material containers), the accumulated quantity is simply the number of entities transported (see Figure 3.3b1). Note that the border flows include components for every material (or scenario) under consideration in both directions separately.

For the computed zonation, a *zone graph* describing the connectivity is created. It is a directed acyclic graph where each vertex corresponds to a zone. For each pair of vertices in this graph, there exists a directed edge connecting them if and only if the two corresponding zones are adjacent. In the computed zone graph, every edge needs to be attributed with the corresponding pair of border flow values (see Figure 3.3c). The first value indicates the material quantity transported along the directed edge, while the second value indicates the material quantity transported in the opposite direction. From this follows that the direction of an edge in the graph is independent from the flow direction. The directed and acyclic properties of the zone graph merely simplify its traversal, but the flow directions follow from the border flow values.

3.6 Spatial Embedding of the Zone Graph

In the previous section, we described a way to generically express material movement in the domain of interest via border flows between adjacent zones. The zone connectivity is represented as an abstract zone graph with border flow values attributed to the edges of this graph. To obtain the geospatial flow map (see Figure 3.3d), vertices and edges of the graph have to be embedded into the spatial domain. The spatial embedding of the abstract zone graph relies on *zone representations*, which are geometric characterizations

Figure 3.4: Overview of spatial embedding. (a) Input zone graph with two flow components (red, grey). (b) Input zone representations. (c) Connection of relevant zone representations. (d) Resampling of lines and creation of geospatial graph. (e) Iterative vertex merging. (f) Layout of individual flow lines. (g) Removal of unnecessary connection circles. (h) Visual styles and rendering.

of the zones. If a zone originates from a line, such as a street zone, then the representation is the actual street line from which the zone was initially derived. Otherwise, the zone representation is a point set, which may contain one point for smaller zones or several points for large or complex zones. This multi-step process is illustrated in Figure 3.4 and is explained in detail in the following.

3.6.1 Geospatial Graph Creation

For each vertex of the abstract zone graph (see Figure 3.4a), there exists a corresponding spatial zone representation, which is either a line or a point set (see Figure 3.4b). The zone representations are initially disconnected and independent from each other. They need to be connected according to the connectivity information of the zone graph (see Figure 3.4c). Two zones represented by adjacent lines can simply be joined at their endpoints. If one or both of the zone representations are point sets, the two closest points of them should be connected. The resulting set of connected lines is resampled with an application-specific step size (logistics: 100 m, water: 10 m, evacuation: 7 m) to unify the density of line points in preparation for a subsequent vertex merging step. Each point of every resampled line is interpreted as a vertex of a new *geospatial graph* (see Figure 3.4d). Connections between the points are interpreted as edges, of which each inherits the flow values from the corresponding edge of the zone graph. The resulting geospatial graph (see Figure 3.4d) has generally more vertices and edges than the original abstract graph as all spatial points of the zone representations are now individual graph vertices. If all components of the flow value assigned to an edge are zero, the edge is removed. Potentially disconnected graphs resulting from this removal do not require any special handling.

3.6.2 Force-Driven Semantic Levels of Detail

The geospatial graph created in the previous step has the highest possible level of detail, and the input flow data is replicated multiple times. This redundancy should be decreased so that distinct flows can be identified more easily. Therefore, we apply a semantic-driven simplification to the geospatial graph. This is done by using our iterative force-driven merge algorithm (see Figure 3.4e) inspired by the work of Debiasi et al. [DSD14]. We introduce attraction and repulsion forces between the geospatial graph vertices. At each iteration, we move the vertices according to these forces. Vertices within a given radius r are then merged to a new vertex at their centroid. The algorithm terminates when the forces between the vertices reach an equilibrium or if a specified maximum number of iterations is exceeded. In our applications, the algorithm always converged in less than 100 iterations.

A main difference of our algorithm to the one by Debiasi et al. is that we do not assume a distinct flow direction from a single root vertex to multiple leaf vertices. Instead, flow can occur in both directions between any adjacent zones, so there is no distinction between source and sink vertices. Debiasi et al. also rely on a parent-child relationship between vertices for the calculation of forces, therefore implicitly ranking their importance.

With bidirectional flows, such hierarchies are not given. Instead, we only use a flag for important vertices that should never be moved or merged, such as material depots in a logistics application.

In our method, an attraction force $\vec{F}_a(v)$ pulls each vertex v from the set of all vertices V towards all other vertices, weighted by their flow magnitudes:

$$\vec{F}_a(v) = \sum_{v_i \in V} \left(\frac{i(v_i) + o(v_i)}{i(v_i) + o(v_i) + i(v) + o(v)} \cdot \frac{\vec{p}(v_i) - \vec{p}(v)}{\|\vec{p}(v_i) - \vec{p}(v)\|^2} \right),$$

where $i(v)$ and $o(v)$ denote the summed (over all incident edges and all flow components) inflow and outflow magnitudes of v, respectively, and $\vec{p}(v)$ is the position of v. A stress force $\vec{F}_s(v)$ moves each vertex v towards a weighted centroid of the set V_N of vertices adjacent to v:

$$\vec{F}_s(v) = \sum_{v_i \in V_N} \left(\frac{f(v, v_i) + f(v_i, v)}{i(v) + o(v)} \cdot \left(\vec{p}(v_i) - \vec{p}(v) \right) \right).$$

Here, $f(v, v_i)$ denotes the summed magnitude of all flow components directed from v to v_i. Additionally, if the set of important vertices $V_I \subseteq V$ is not empty, we apply a repulsion force $\vec{F}_r(v)$ to all other vertices $v \in V \backslash V_I$ to segregate the important vertices from all other vertices in the flow map:

$$\vec{F}_r(v) = \sum_{v_i \in V_I} \frac{\vec{p}(v) - \vec{p}(v_i)}{\|\vec{p}(v) - \vec{p}(v_i)\|^2}$$

All these forces are applied to each vertex $v \in V \backslash V_I$ via a displacement vector $\vec{D}(v)$ added to the position of v:

$$\vec{D}(v) = k_a \cdot \vec{F}_a(v) + k_s \cdot \vec{F}_s(v) + k_r \cdot \vec{F}_r(v),$$

where k_a, k_s, $k_r \in [0, 1]$ are weights for the forces to control the convergence of the algorithm. After displacement, all vertices closer to each other than a radius r are merged into a new vertex placed at the centroid of their locations.

The second major difference to the work by Debiasi et al. is that neither the force weights k_a, k_s, k_r nor the merge radius r are constant. Instead, only a range is defined for these parameters, but the actual values within the range are location-dependent. The ranges have been found empirically for the different applications (logistics: $r \in [70, 1400]$ m, $k_a \in [0.1, 1]$, $k_s \in [0, 0.05]$, $k_r \in [0, 1]$, water: $r \in [1, 100]$ m, $k_a = k_s = k_r = 0$, evacuation: $r = 10$ m, $k_a = k_s = k_r = 0$). The values at point $\vec{p}$ in the spatial domain are determined by blending the maximum and minimum of the corresponding ranges with a blend weight, which is the level of detail $\lambda(\vec{p}) \in [0, 1]$. A higher $\lambda(\vec{p})$ leads to smaller forces and to less merging of vertices. This preserves details of the zone representations (for example of street lines) and thus leads to a stronger correspondence of the resulting flow map with the geospatial context. A lower $\lambda(\vec{p})$ leads to a more generalized flow map in which border flows close together are merged.
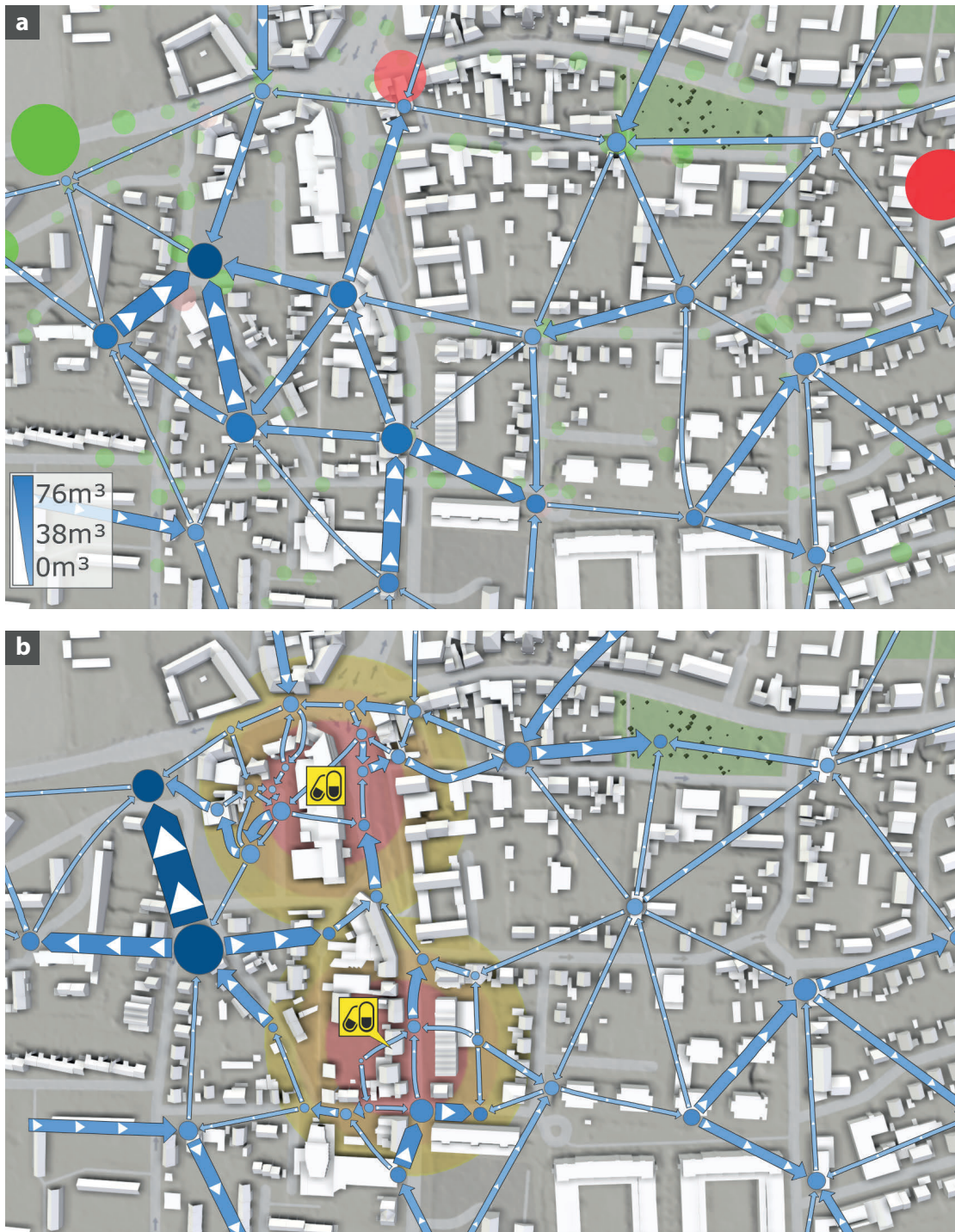
Figure 3.5: Flow maps for heavy rain scenarios. (a) Overview. Circles indicate the sewer outflows (red) and inflows (green). (b) Locally increased level of detail nearby important buildings.
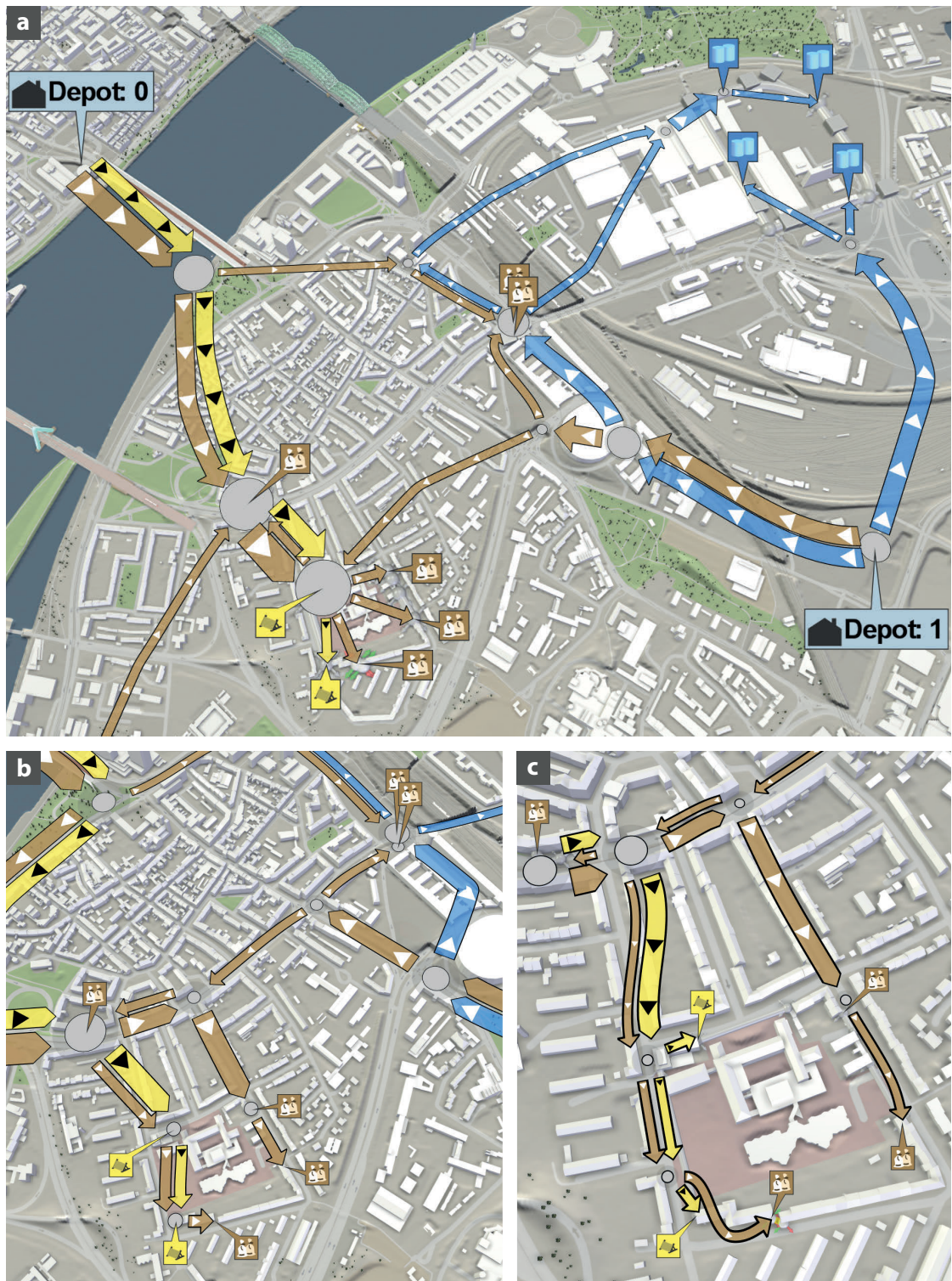
Figure 3.6: Zoom-dependent level of detail. (a) Low. (b) Medium. (c) High.

We choose $\lambda(\vec{p}) = \max(\lambda_s(\vec{p}), \lambda_v(\vec{p}))$, where $\lambda_s(\vec{p})$ and $\lambda_v(\vec{p})$ are the semantic and view-dependent levels of detail, respectively. Namely, $\lambda_s(\vec{p})$ assigns an importance value in $[0, 1]$ to each point $\vec{p}$ in the spatial domain. We use the semantic level of detail to reduce the visual complexity in context regions while emphasizing focus regions like the neighborhood of important infrastructure, as shown in Figure 3.5 for two pharmacies. A view-dependent level of detail $\lambda_v(\vec{p})$ allows for the interactive control of the granularity of the flow map. It is dependent on the distance between the view point and $\vec{p}$, which is normalized to $[0, 1]$ using an application-specific maximum distance (logistics: 4000 m, water: 400 m, not used in evacuation). Therefore, an overview flow map is automatically refined as the user zooms in on an area of interest, while keeping the density of flow lines displayed on the screen approximately constant. In Figure 3.6a, an overview flow map for material deliveries is shown. As the user zooms into the hospital area, the flow map is gradually refined (see Figure 3.6b,c).

3.6.3 Layout of Composite Flow Lines

The result of the above algorithm is a geospatial graph simplified according to the required level of detail. To visualize this graph, for each edge, lines have to be created for all flow components in both directions. These lines should be visually connected at their end vertices without overlaps. In contrast to existing uni-directional flow map algorithms, we cannot simply join the individual lines and visualize junctions as arborizations without giving a false impression of a principal flow direction. Arborizations would also introduce unavoidable overlaps increasing exponentially with the number of flow components. We solve this by using connection circles at places where multiple adjacent lines connect.

At each vertex position, a circle is created that needs to be large enough to fit the widths of all incident edges on its circumference. Let E_c be the set of all edges to be connected to a circle c. Each edge $e \in E_c$ is essentially a group of individual lines for each separate flow component and direction. A user-defined transfer function τ is used to compute the width of a line l from its flow magnitude m_l. Since the width w_e of the entire edge is the sum of the widths of all lines $l \in L_e$ belonging to it, the required radius of the circle c should be:

$$r_c = \frac{1}{2\pi} \cdot \sum_{e \in E_c} w_e = \frac{1}{2\pi} \cdot \sum_{e \in E_c} \sum_{l \in L_e} \tau(m_l)$$

In practice, this is not sufficient. If the incoming directions of all edges are approximately the same, half of the edges have to be connected to the part of the circle facing away from them, leading to unnecessary stretching. Doubling r_c solves this problem, but usually leads to unnecessarily large circles. Empirically, we found that scaling r_c by a value between 1.3 and 1.5 gives sufficiently good results.

Let e be an edge with a normalized direction $\vec{d} = (d_x, d_y)^T$ that needs to be connected to the connection circles c_a and c_b at its start and end, respectively. Then $\alpha = atan2(d_y, d_x)$ is the polar angle of a point on the circumference of c_a to which e should be connected. The polar angle for connecting e to c_b is $\beta = atan2(-d_y, -d_x)$. These angles lead to
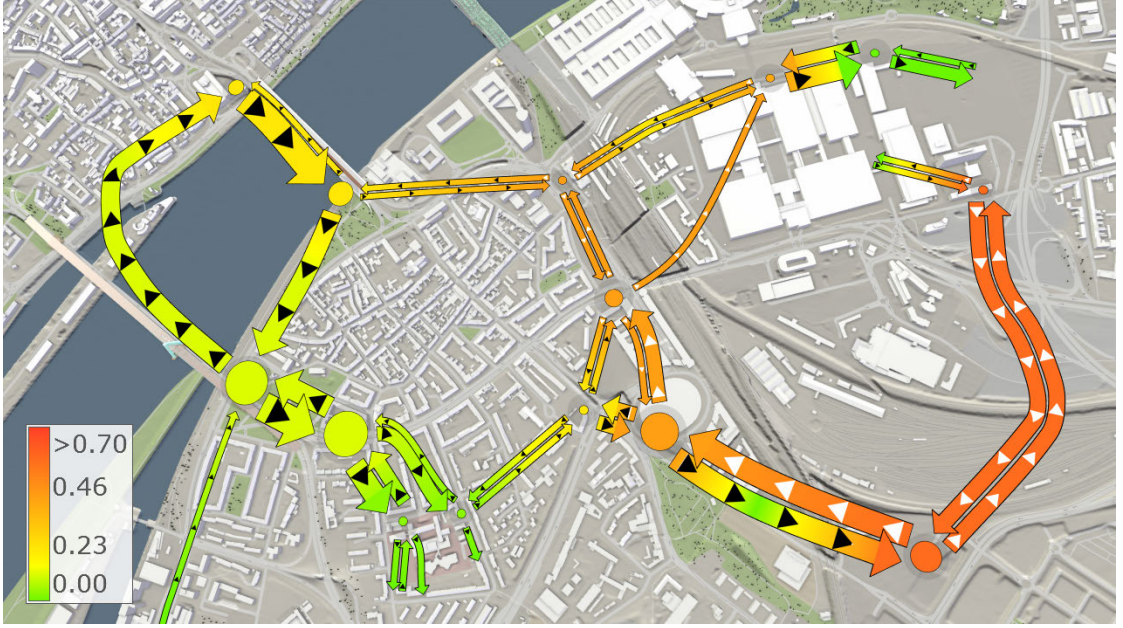
Figure 3.7: Overlay visualization of traffic conditions aggregated over a time span.

the shortest line between the two corresponding connection circles. However, this is not generally optimal. As all edges have widths, using only direct connections can lead to overlaps for edges with similar directions. To avoid overlaps, a relaxation step is applied to each circle which changes the connection angles of all incident edges to fit them on the circumference. Since the circumference is greater than or equal to the sum of the widths of all edges, an arrangement without overlaps is guaranteed to exist. The optimal arrangement is free of overlaps and has the smallest summed difference of new polar angles $\hat{\alpha}$, $\hat{\beta}$ to the original polar angles α, β, i.e., it has the shortest connections possible without overlaps. Solving this optimization problem with the time constraints of an interactive visualization is very challenging. We approximate the solution by generating different connection arrangements and choosing the one with the smallest sum of connection lengths. To generate one arrangement, the first edge is connected directly to the connection circle. All subsequent edges are checked for overlaps with already connected edges. If an overlap exists, the current edge is moved on the circumference in both directions until it fits in a gap without overlaps. The different arrangements are generated by permuting the order in which the edges are inserted. We generate a maximum of $6! = 720$ permutations, which allows the flow map generation to be fully interactive while producing near-optimal connection arrangements for vertices with up to six adjacent edges.

The new start point of edge e on the circumference of the connection circle c_a is determined by adding the radius-vector $\vec{v}_a = r_a \cdot (\cos(\hat{\alpha}), \sin(\hat{\alpha}))^T$ to the center of c_a. The respective radius-vector for connection circle c_b of the end point is given by $\vec{v}_b = r_b \cdot \left(\cos(\hat{\beta}), \sin(\hat{\beta})\right)^T$.

Figure 3.8: Overlay visualization of road inaccessibility due to inundation.

However, these connection points have been calculated for the edge as a whole, which is now split up into individual lines for the flow components. These lines should be offset relative to the connection point according to their widths. The offsets are computed along the edge normal so that all lines belonging to the same edge remain parallel (see Figure 3.4f). There are different options to group and order the individual lines. For our applications, we found it most useful to group them by flow component or by direction. In the latter case, we apply the largest offset in the direction normal to the edge to the lines with flow in edge direction. This way, flow in edge direction is displayed on the right, mimicking the rules of right-hand traffic. The normalized radius-vectors $\vec{v}_a$ and $\vec{v}_b$ are used as connection tangents for a Hermite spline approximation of the individual lines. This allows the lines from edges that have been offset significantly in the relaxation step to connect to the circles more smoothly.

So far, all vertices are represented by circles. This is only necessary for vertices with three or more incident edges. The complexity of the flow map can be reduced by removing unnecessary circles (see Figure 3.4g). If a vertex has only one incident edge, all individual lines are now extended until the middle of the removed circle. If a vertex has two incident edges and the flow values of both edges are equal, the corresponding individual lines of both edges can simply be connected.

3.6.4 Rendering

At this point, we have a collection of smooth lines corresponding to individual flow components. For rendering a line, we generate a triangular mesh ribbon from the points

Figure 3.9: Overlay visualization of wave arrival times [hh:mm] after a floodwall breach.

of this line. As stated above, the width of the ribbon for a line l is given by the mapped flow magnitude $\tau(m_l)$. An arrow tip is added at the end of each ribbon to visually express the direction of the flow. The arrow tip can either be a simple narrowing of the line width to a single point or an extruded arrow head. The latter method is used for lines with a screen-space width below a specified threshold.

Coloring is applied to the ribbon, which can either be a solid color distinctive for individual flow components (see Figure 3.1a) or some value mapped to a color with a user-defined transfer function. A natural choice for this value is the flow magnitude m_l corresponding to the line l (see Figure 3.3d). It is also possible to add new geospatial information to the flow map as an overlay. For example, in applications involving a traffic network, visualizing the traffic conditions (see Figure 3.7) or accessibility of roads (see Figure 3.8) with color can help to highlight problematic areas. In Figure 3.9, we visualize the wave arrival times in a breach scenario as an overlay. In Figure 3.1b, a density measure of the evacuating crowd is shown with the color overlay. For such applications, we evaluate the mapped property for each vertex of the original geospatial graph before applying the iterative vertex merging and then keep track of the values while merging. This allows us to visualize aggregates such as the minimum, average, and maximum road accessibility of each vertex of the merged geospatial graph. On top of the flow ribbons, glyphs are added. We limit ourselves to repeating directional arrows to indicate the flow direction. They are either black or white, depending on which gives a better contrast to the color of the arrow (see Figure 3.3d). Depending on the application, more complex glyphs can be used.
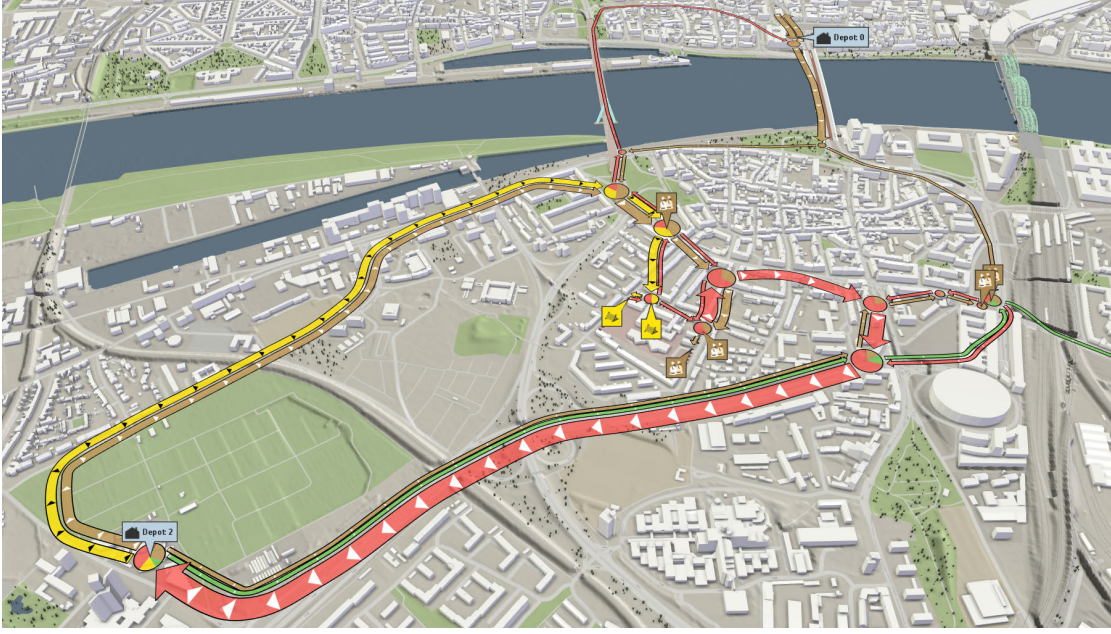
Figure 3.10: Delivery routes by material type. Connection circles visualize the ratios of the different materials in pie charts.

Finally, circles are rendered at the spatial positions of vertices with three or more incident edges. The radius r_c^m of each rendered circle c should directly correspond to the flow magnitude through the corresponding vertex. It is chosen as

$$r_c^m = \tau \left(\sum_{l \in L_c} m_l \right),$$

where L_c is the set of all lines meeting at this circle, m_l is the magnitude of the flow component represented by the line l, and τ is the mapping introduced above. In general, r_c^m is not equal to the radius r_c of the connection circle, so the connection circle of radius r_c does not properly represent the flow through the corresponding vertex. It is rather a circle to which the lines connect. We show both circles at the spatial position of the vertex, but give the larger connection circle just enough opacity to be barely visible (see Figure 3.3d1). This provides a hint to which vertex an edge connects. The opaque inner circles allow for an accurate comparison of flow magnitudes by size. To these circles, the same coloring as to the ribbons can be applied. Alternatively, a pie chart can be displayed which shows the ratios of different flows through this vertex (see Figure 3.10). These pie charts can also be assigned a maximum capacity to visualize the utilization of important infrastructure, for example a material depot in a logistics application.
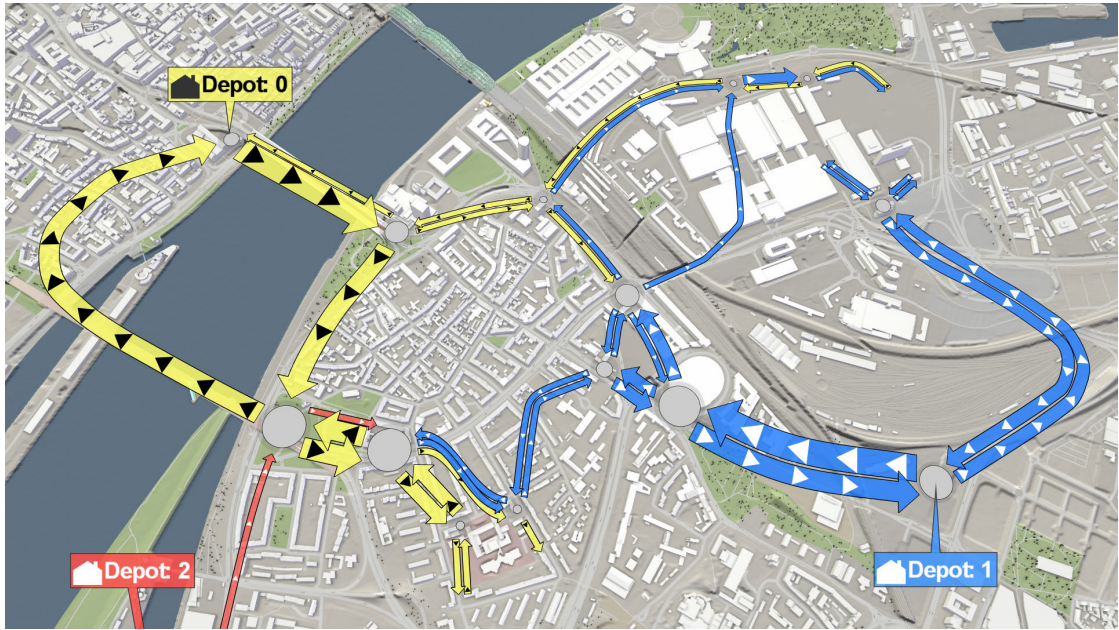
Figure 3.11: Flow map composition according to different material origins, i.e., construction material depots in the logistics application.

3.7   Results and Evaluation

We implemented our technique in Visdom and demonstrate it in three different applications. The logistic application considers planning of delivery routes in an urban area. In the field of flood management, we address the surface water movement and interaction with the sewer network in an urban area at the time of a heavy rain. The evacuation application considers evacuation scenarios for an office space. All applications feature basic navigation for view-dependent levels of detail, interactive flow decomposition, and animation of the flow maps over time. For a demonstration of these interactive features, we refer to the video accompanying the original paper [Com].

The evaluation consists of two phases conducted before and after the implementation. The involved committee includes a logistics expert, an expert for sewer networks, two consulting engineers for integrated catastrophe management, an expert for hydrology, and an expert for crowd modeling. During a preliminary evaluation, the experts gave feedback on mockups of the planned results and provided us with hints on actual problems in their fields for which composite flow maps could be helpful. For the evaluation of the final results, the experts were shown composite flow maps for the different applications and were asked if they found them expressive, helpful, and aesthetic. In general, the experts concurred that our composite flow maps are an effective and aesthetic way to present different material flows, including quantities, in the geospatial context. In particular, flow maps provide a comprehensible way to get a quick overview of time-dependent flow

Figure 3.12: Interactive flow map decomposition according to rooms selected by the user (yellow). (a) Combined with all other rooms (blue). (b) Flows from a different set of selected rooms.

data for logistics and evacuation applications. We received mixed feedback regarding the heavy rain application. Visualizing major flows of heavy rains is a fairly new yet important task for flood management. According to the domain experts, our method is a step in the right direction but needs further improvements.

Composite flow maps in the logistics application consist of separate flow components for the deliveries of different materials (see Figure 3.1a) and the deliveries from different depots (see Figure 3.11). For the evacuation, individual rooms can be selected interactively

65
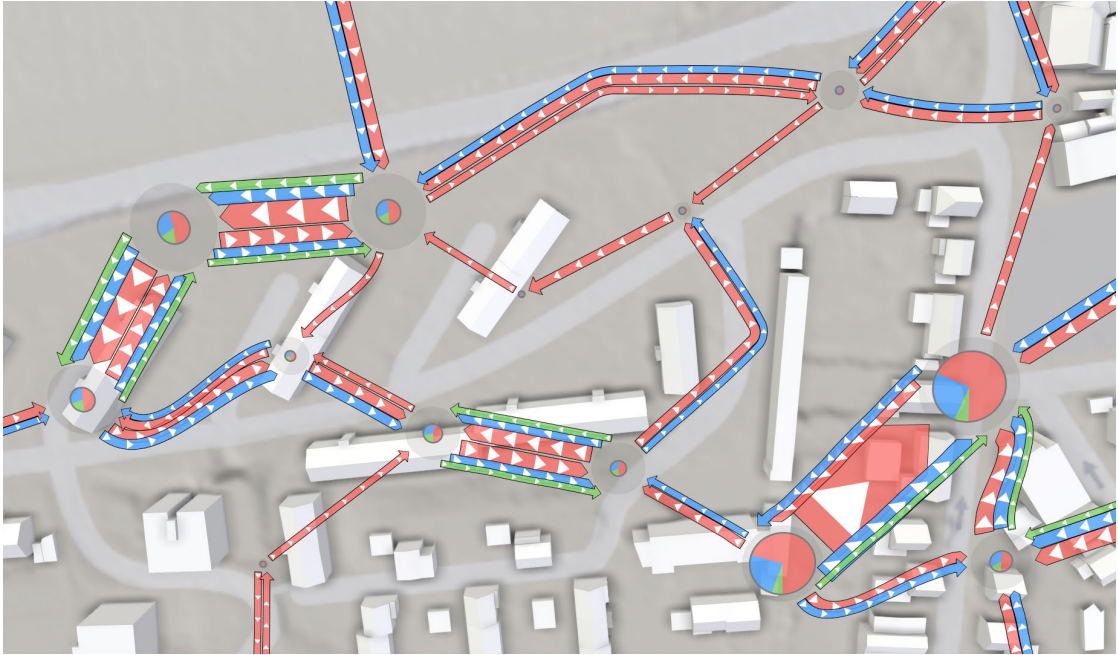
Figure 3.13: Uncertainty in water transportation for a heavy rain simulation. Different flow components correspond to different aggregations of ensemble simulations, namely minimum (green), average (blue), and maximum (red) water flows.

to separate the crowd flow originating from these rooms (see Figure 3.12) from the rest of the flow. This was appreciated by the domain experts as a helpful feature for planning. We also evaluated the use of flow maps for uncertain data in the context of heavy rain simulations. A flow uncertainty visualization over multiple different flooding scenarios, where the components show the minimum, average, and maximum flows (see Figure 3.13), was seen as useful only for very specific tasks such as debris transport by flood water. The zoom-dependent levels of detail (see Figure 3.6) were seen as a helpful feature in both the heavy rain and the logistics applications.

In Figure 3.5a, an overview flow map for flooding by heavy rain is supplemented with circular glyphs for the sewers that consume (green) or emit (red) significant water amounts. This was welcomed by all domain experts. In Figure 3.5b, we use a semantic level of detail based on the proximity to pharmacies for keeping the flow representation in the surrounding area more accurate. This was well received by the experts. However, for this application, more generalized flow maps were considered misleading, because the visible relation to the actual zones is lost. This might improve with an enhanced zonation algorithm. On the highest level of detail, we display an arrow glyph representation of the sewer inflows and outflows using the same visual encoding as for the flow map (see Figure 3.14). This was considered a good visualization of overflows for planning tasks.

Figure 3.14: Glyphs show details about the sewer-surface coupling.

We visualize additional geospatial information on top of the flow map if necessary. In the logistics application, these are traffic conditions obtained from a public service (see Figure 3.7) or the inundation of streets (see Figure 3.8). This highlighting of potential transportation bottlenecks based on dynamic data was highly rated for planning applications and characterized as superior to current systems. In the heavy rain application, we visualize the wave arrival times to encode the temporal dimension of the scenario in the flow map (see Figure 3.9). For evacuation, we display the occupancy of rooms over time (see Figure 3.1b) to highlight critical sections on the evacuation routes. In total, overlay visualizations for all three applications have been highly appreciated. However, our approach to always use connection circles in the flow map layout has received mixed feedback due to possible misinterpretation. Especially for the evacuation scenarios, it is desirable to use arborizations where possible. Displaying pie charts on top of connection circles (see Figure 3.10) was considered useful for determining the workload at a particular junction and for easy comparison of the quantities of different flow components. Finally, animating the flow map over time was considered especially helpful for the logistics and evacuation applications, as it allows the user to investigate the chronology of events and shows how critical regions emerge in the evacuation scenario.

In summary, our main contribution—the composite flow map—has been highly rated by all six experts, the interactive decomposition of flows was considered useful by five experts. The use of composite flow maps was seen as an improvement for their work in two out of three applications. Levels of detail used in the logistics and water application were appreciated by four out of five experts. All six experts concurred that the overlay

visualization of additional data on top of the flow maps was very helpful for common tasks in their fields.

3.8   Conclusions and Future Work

In this chapter, we propose a simplification and visualization technique for material movement data. It fills the gap between conventional flow maps for rooted trees and edge-bundling for arbitrary multigraphs. With composite flow maps, multiple flows can be shown in a single visualization. The visualization can be simplified with interactive and locally varying levels of detail while preserving the representative magnitudes of the generalized flows. Using the layout algorithm, visual representations of flow components are arranged in the geospatial domain without overlaps.

Our technique relies on a unified representation of flow data by means of a weighted zone graph, which only requires a flow of material in one or two dimensions between zones. The required zonation of the spatial domain can range from a regular subdivision to complex semantic structures. This allows for the generic treatment of flows for different applications and data modalities. We demonstrate the wide applicability of our technique with three different types of flows. Other possible applications include the visualization of computational fluid dynamics data (for example wind, gas flows), particle-based simulations, and also non-spatial transportation data such as network traffic.

The evaluation partners from various professional fields emphasize the benefits of our technique for planning in their applications. There is a demand for techniques allowing the user to investigate and directly compare entire flows as well as parts of the flows isolated in separate flow components. This is especially relevant for applications in the fields of logistics and evacuation planning.

For the heavy rain application, the domain experts found it difficult to draw conclusive information from our visualization. They attribute this to the diffuseness of the original movement data. Reliable extraction and visualization of principal flows in heavy rain simulation data remains a challenging task. Further research is needed to investigate whether this can be accomplished by considering the data themselves in the zonation process to align zones with existing flow trends. Existing work on the simplification and hierarchical representation of vector fields [TvW99, JRS12] might serve as a good starting point for this task.

Another direction of future work is the use of composite flow maps to visualize the uncertain transport of sediments, debris, and pollutants in water. A need for these applications is expressed by the experts. One more goal is the use of arborizations at junctions in the layout algorithm. A hybrid approach using arborizations in unidirectional segments of the flow can further reduce the visual complexity.

Interactive Flood Visualization

4.1 Introduction

The ubiquitous use of computer simulations in flood and stormwater management creates a growing demand for expressive and efficient techniques to visualize the corresponding, often very large flood-related data (see Figure 4.1). As interactivity is a key requirement for modern decision support tools, the performance of simulations becomes a crucial aspect. The results of the fast grid-based shallow water simulation using the finite-volume method are discretely defined on rectangular grids in the form of height fields. Visualization of such height fields requires some sort of surface reconstruction by means of interpolation or approximation. There is a trade-off between the computational cost of the interpolation and its visual appearance. The simplest form of interpolation, nearest neighbor, produces results that are far from being physically realistic (see Figure 4.2a), which makes it practically useless for engineering or presentation purposes. Bilinear interpolation introduces C^0-continuity, but leads to perfectly straight or angular shoreline features and discontinuities in shading (see Figure 4.2b). A higher order of continuity is needed for a better visual appearance. For regular grids, C^1-continuity is achieved by bicubic interpolation (see Figure 4.2c) or approximation (see Figure 4.2d), which are fast enough for interactive applications.

To keep up with the increasing scale of simulated scenarios and the generated data, adaptive grids can be used. However, the application of adaptive grids introduces additional challenges for visualization. Currently, one has to choose between fast linear interpolation [KT09, Lia11] and offline smooth interpolation [BMA10, BOR14, GJAG14, FBHD17]. None of these algorithms provides a continuously differentiable surface reconstruction while also being efficient enough for interactive applications. The main challenge of adaptive grid interpolation is the absence of implicitly given relationships between neighboring cells of different resolution. Within a neighborhood required for third-order interpolation, a potentially infinite number of alternative cell arrangements is possible. Retrieving the

69
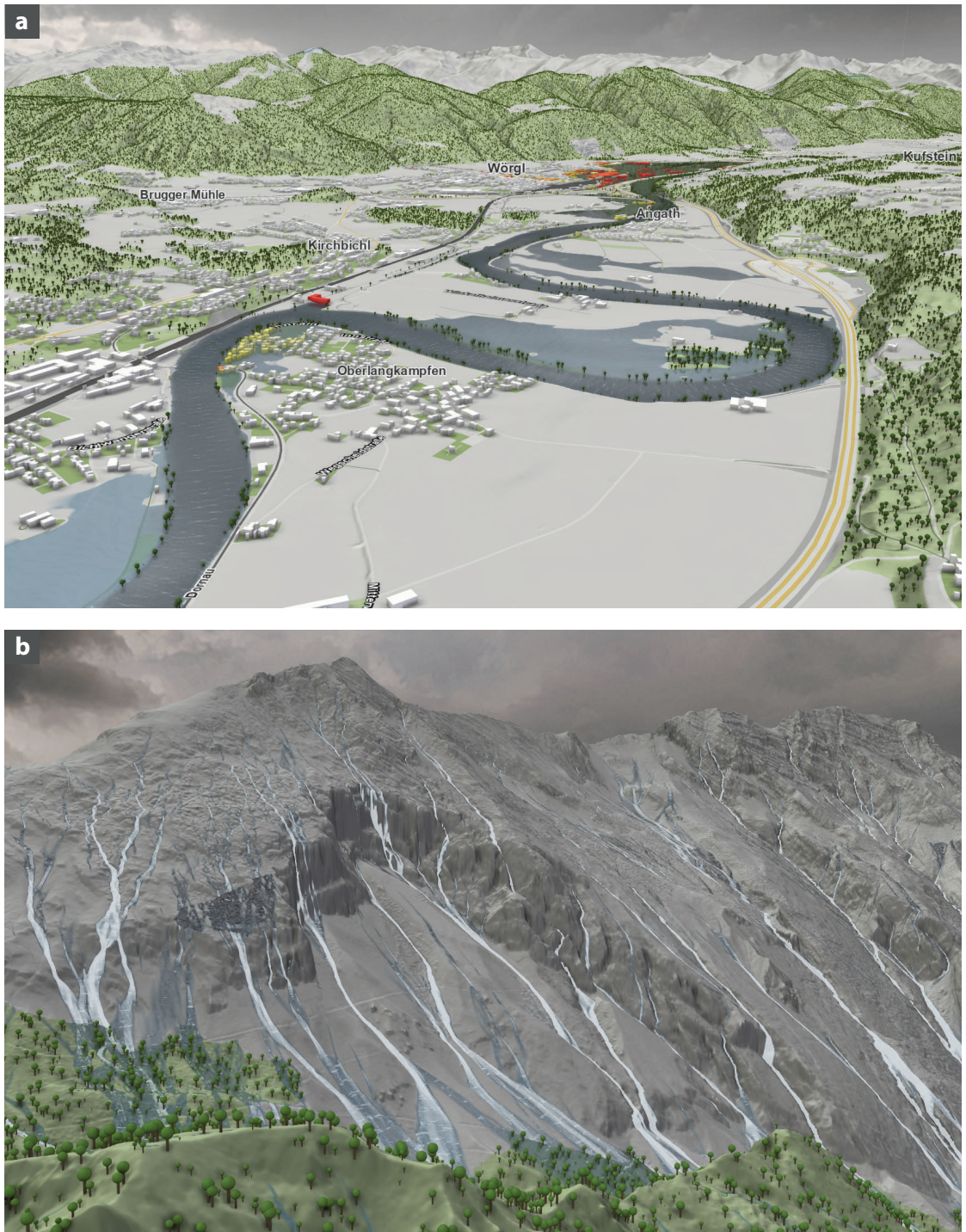
Figure 4.1: Visualization of smoothly interpolated flood simulation results and terrain defined on adaptive grids. (a) Large-scale flooding of villages by a nearby river. (b) Stormwater runoff in a mountainous region. High velocities are indicated by white foam.
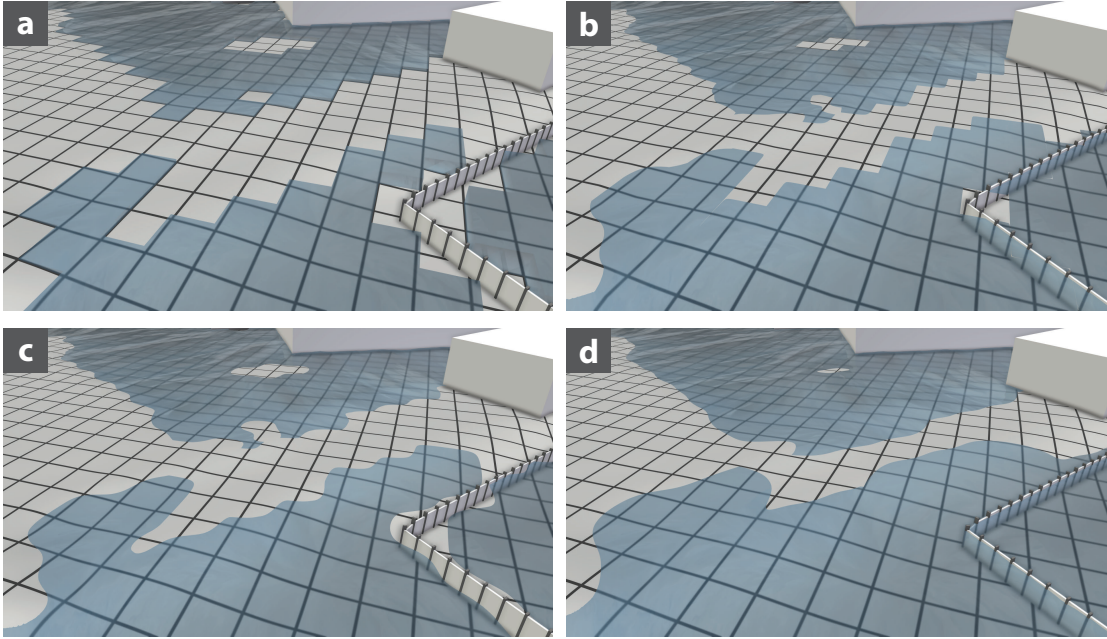
Figure 4.2: Surface reconstruction with interpolation. (a) Nearest-neighbor and (b) bilinear interpolation lead to linear and angular shoreline features. A smoother shoreline is obtained by bicubic interpolation with (c) Catmull-Rom splines and (d) cubic B-splines.

neighborhood values is computationally hard, yet necessary to achieve smooth surface transitions between grid cells of different resolution. One contribution presented in this chapter is a novel method for fast C^1-continuous third-order interpolation on adaptive grids. Trivially, the presented method can also be reduced to a C^0-continuous linear interpolation.

Using a fast, but simple interpolator for surface reconstruction, such as usual bilinear or bicubic interpolation, the reconstructed water surface is not always plausible. This is particularly apparent at dry/wet boundaries. Here, two types of features can manifest, which we consider artifacts in the context of water visualization. First, if the height field is not defined everywhere, missing values at dry cells have to be deduced from existing data. Previous work [HWP+15, HPW+16] used the terrain elevation as water level at these dry cells, which can lead to uphill *climbing* of water (see Figure 4.3a1) as well as *diving* towards walls (see Figure 4.3b1). Second, wall boundaries are discretized on the rather coarse grid used for simulation and interpolation. Therefore, the interpolated extents of simulated inundation do not always match the detailed wall geometry used for visualization, leading to seeming *leaking* (see Figure 4.3c1). During live sessions in the past, domain experts repeatedly misinterpreted these artifacts as actual water propagation. We present a surface reconstruction for water height fields with boundary conditions that reduces these artifacts (see Figure 4.3, right column) in both regular and adaptive grids.
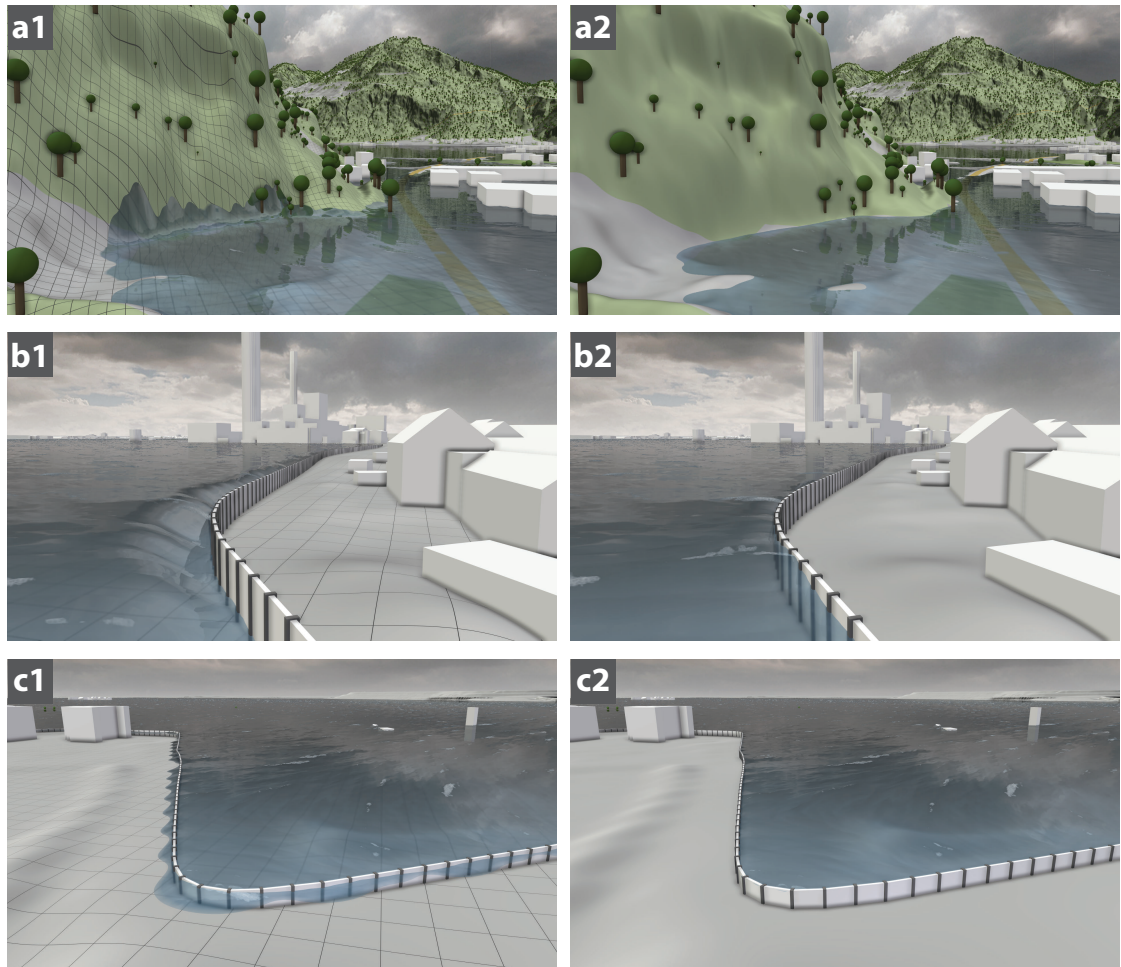
Figure 4.3: Common artifacts with standard interpolation (left column) compared to our results (right column). (a) Climbing. (b) Diving. (c) Leaking.

Another challenge tackled in this chapter lies in the wide range of levels of detail that has to be offered by height field rendering for geospatial visualization. A typical task in flood management requires modeling of floods over large areas such as a major city or multiple smaller communities (see Figure 4.1a). At the same time, protection measures have to be tested interactively at a scale of smaller regions or individual buildings. This requires frequent zooming from an overview of an entire country or region showing millions of cells to a close-up of individual one-meter cells. We present a simple view-dependent method for recursive tessellation on the GPU to provide a continuous level of detail for large height fields.

The correct perception of flow behavior is important for the interpretation of water simulation results. However, it is hard to extract this information from a colored surface alone. The use of glyphs or other established flow visualization techniques could solve

the problem, but would result in a cluttered visualization that attracts all the viewer's attention. In nature, flow directions can often be identified easily by waves and patches of foam flowing on the water surface. We demonstrate the use of exaggerated waves and foam derived from simulation data as visual metaphors for the intuitive indication of flow behavior (see Figure 4.1b). Here, the challenge is to use simulation results for a direct derivation of foam and waves that give a realistic and intuitive indication of the water depths and movement patterns.

Although water rendering for entertainment purposes, for example for video games, is a well-researched topic, the interactive visualization of time-dependent flood simulation data is still a challenge. In previous work, the complex problem of water rendering has been simplified based on assumptions that certain data do not change. However, such assumptions do not hold for dynamic simulation data. For example, water bodies in video games are usually confined to a static triangle mesh or plane such that the water surface can be perfectly aligned to the scene geometry by an artist manually. This alleviates the need for expensive interpolation and dealing with artifacts, but prohibits the dynamic inundation of the scene. Likewise, using precomputed flow data for surface shading is sufficient for an aesthetic appearance. For the accurate visualization of flows in a dynamic velocity field, this simplification makes no sense. Therefore, we feel the need to set ourselves apart from these techniques which are limiting in our case. Instead, we introduce more suitable techniques tailored to our particular application.

In summary, the scientific contributions presented in this chapter are:

- Continuous third-order and linear interpolation methods for height fields defined on adaptive grids

- A surface reconstruction for water height fields including correct treatment of boundaries to reduce reconstruction artifacts

- A view-dependent recursive tessellation algorithm for visualization of height fields with continuous level of detail

- Water shading with waves and foam derived from simulation data for better depth and flow perception

The described techniques for the reduction of artifacts and for water surface shading are independent from the simulation grid and can be used for adaptive and regular grids alike. We have thoroughly evaluated the presented results in live sessions and with a user survey among 96 participants consisting of experts in the field of flood management as well as members of the general public. Benchmarks demonstrate the high performance of the proposed techniques in real-world scenarios.

4.2 Related Work

Interpolation Fast interpolation of data defined on adaptive grids is of great interest in volume rendering, where data sets become large rapidly. Here, the common practice of interpolation is to map the data to multi-level textures and then facilitate trilinear hardware interpolation [WWH$^+$00, KH02, BHMF08]. In applications that also require gradients, the linearly interpolated values of central differencing can be reused for third-order reconstruction without additional cost [Csé19]. For adaptive-mesh-refinement grids, the stitching method is commonly used, which inserts special stitch cells with pre-defined interpolation behavior at level transitions [WKL$^+$01, ME11, BST15]. For interpolation within quadtrees, Kim and Tsiotras [KT09] propose a bilinear interpolation scheme that uses bilinearly upsampled values in transition regions between levels. A similar scheme proposed by Liang [Lia11] uses an enclosing triangle for upsampling. All of these methods use a linear interpolator on the upsampled regions and thus yield C^0-continuous results. However, an error assessment by Kidner [Kid03] shows that higher-order interpolation of digital elevation models leads to a significantly more accurate surface representation than linear interpolation. The scheme proposed by Min and Gibou [MG06] uses a biquadratic interpolator, but does not consider continuous differentiability in transition regions.

C^1- or C^2-continuous local refinable splines defined on hierarchical data structures are often used for isogeometric analysis [DCL$^+$08, BLE$^+$14, LCKD16]. Fuchs et al. [FBHD17] use them for volume rendering and report runtimes of 38 milliseconds for around 256 000 cells. For unstructured data, interpolation with radial basis functions such as thin plate splines can be used. Hutchinson [Hut95] demonstrates an interpolation of rainfall data, resulting in a C^1-continuous surface. Smooth interpolation of scattered data can be accelerated by combining locally defined thin plate splines [Fra82]. With an evaluation of thin plate splines on the GPU, Beatson et al. [BOR14] report a runtime of 1.3 seconds for 128 000 data points. Kriging, a technique for geostatistical prediction, has also been used for the approximation of scattered geospatial data [Goo00, Ree00]. With GPU-based implementations, the processing of a few thousand data values takes several seconds [Che13, GJAG14]. Natural-neighbor interpolation for unstructured data yields C^1- or C^2-continuous surfaces [Bob08]. With an implementation optimized for the GPU, Beutel et al. [BMA10] report a runtime of 163 seconds for 186 million cells.

To our knowledge, there exists no continuously differentiable interpolation technique that can process adaptive data in the order of one hundred million cells interactively, i.e., in less than 30 milliseconds. Currently, this efficiency is only achieved by grid-based techniques such as regular bilinear or bicubic interpolation. However, the existing approaches are either not suited for adaptive grids or not continuously differentiable, which are given requirements in our application. Our proposed technique fills this gap by extending bicubic interpolation to adaptive grids with very little overhead.

Level of detail The interactive visualization of large-scale height fields is often addressed by level-of-detail approaches. These include geometry clipmaps [LH04], persistent grids [LSGE08], and projected meshes [LNL09]. A comprehensive survey on software

tessellation approaches is provided by Pajarola and Gobbetti [PG07]. Software tessellation approaches running on the GPU achieve a high performance by the use of geometry shaders [HSH10, RRPC12]. Recent approaches exploit hardware tessellation supported by modern GPUs [TBB09, Bon11, KJCH15, NKF$^+$16]. Lee et al. [LJL13] propose recursive tessellation to circumvent limits of hardware tessellation, on which we base our solution.

Water visualization The challenge of visualizing water bodies affects a variety of fields from engineering simulations to the entertainment industry. Often, the water visualization should not only be recognizable [KC14], but should also convey important characteristics such as depth, flow direction, or flow speed. For the realistic appearance of virtual water surfaces, animated waves can be generated by the superposition of wave functions [Tes01, Fin04, NSB13], wavefront tracking [JW15], or by particle-based simulations [YHK07, YNBH09, JW17]. Tile-based approaches map directional features onto surfaces using flow fields [Gri11, vH11, GH12]. Vlachos demonstrates their benefit for route guidance in a video game [Vla10]. Local flow behavior can be visualized using spray particles and foam [BSW10, CM10, DB12, KLCK17]. The perception of water depth can be influenced by appropriate shading methods [PA01, Bel04]. Combinations of the above methods are widely used in video games to increase the degree of realism [StA13, Bow13, Gon16]. Visualizing simulation-based flow information with waves and foam was evaluated for non-professionals [GSH$^+$15] and was judged intuitive and useful.

4.3 Adaptive Grid Data Structure

We implement our findings in Visdom, where we use our proposed approach for the interactive 3D visualization of dynamic water simulation and terrain data. The water flow is simulated with the existing, integrated GPU-based shallow water simulator [HPW$^+$16] using the finite-volume method. The output of the simulation, which serves as an input for our technique, is a collection of scalar and vector fields defined on a rectangular adaptive grid. These data fields include absolute *water levels*, which are water elevations relative to sea level, *water depths*, which are water elevations relative to the terrain, and 2D water velocity vectors. An additional terrain field represents ground elevations relative to sea level. In our system, all discretized data are cell-registered, meaning that data values are associated with cell centers. However, a characteristic of the finite-volume method is that the resulting simulation data are average values over the entire cell rather than exact measurements at the cell center.

For the adaptive grid, we use an extension of the data structure proposed by Liang [Lia11], which is discussed in detail in the following. However, the techniques we present are largely independent of the used adaptive grid. Any grid-based data structure will suffice, provided that it facilitates a quick retrieval of the data value for the cell covering a given 2D position. The only necessary restriction is that neighboring cells, i.e., cells that share a vertex, must not differ by more than one level.
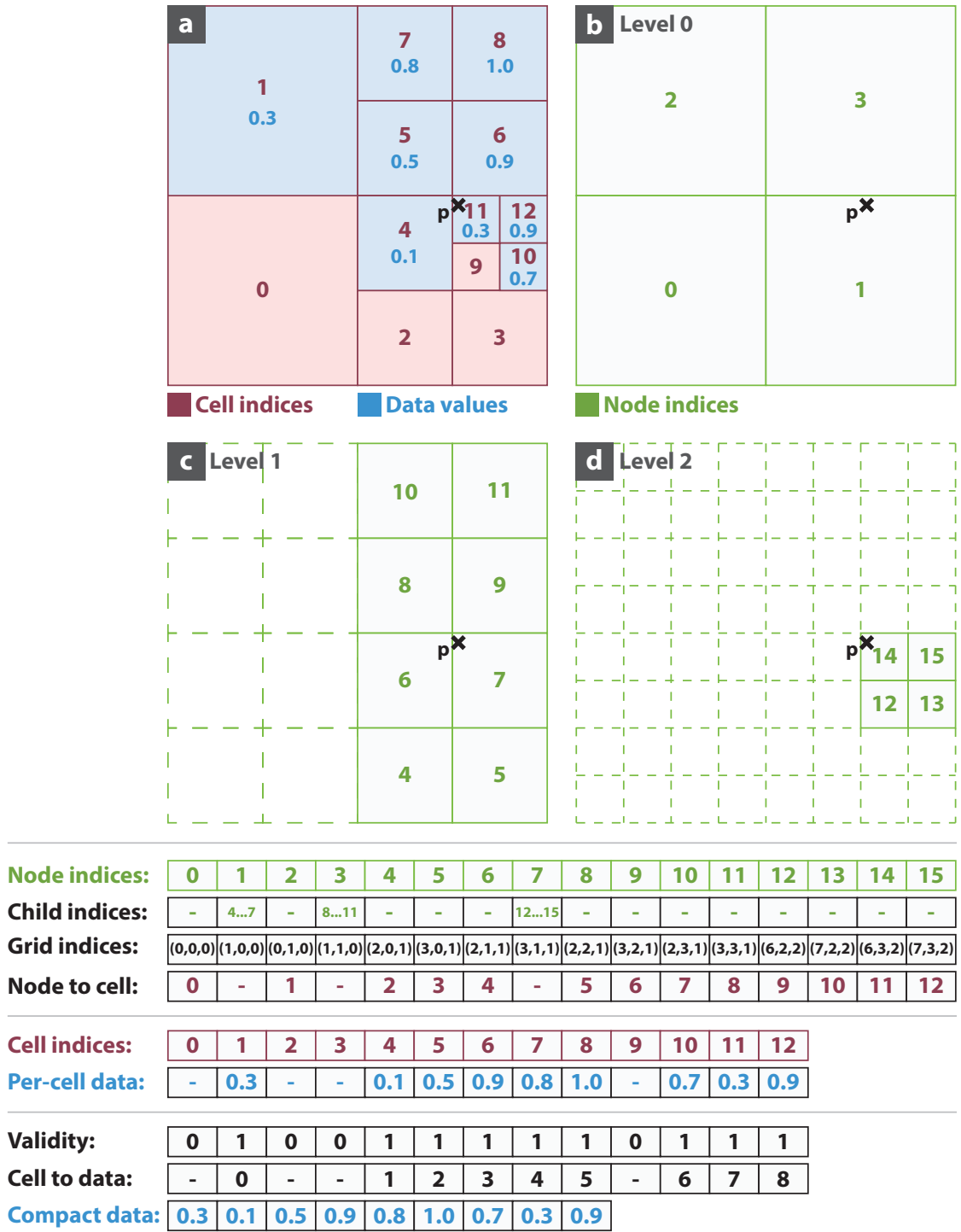
Figure 4.4: Example of the used adaptive grid data structure. (a) Adaptive grid with cell indices (red) and data values (blue). Invalid cells are shaded in red. (b)–(d) Individual levels. (Bottom) Array representation of the adaptive grid and data in Visdom.

The requirements for our application are given by very diverse use cases that the system needs to handle. For example, the modeling of heavy rains considers a region of manageable extents, but requires cell sizes of at most one meter. In this case, almost every cell is wet, i.e., covered by water in the simulation. Another use case is the modeling of large-scale river flooding, for which adaptive simulation grids with tens of millions of cells ranging from one to more than one hundred meters are used. Here, only a small number of cells is actually wet. In urban flood management, interactivity is a key requirement, which is why the simulation grid is rather coarse with a typical cell size of five meters to allow for a fast initial simulation. In all use cases, fast update and display of simulation data are crucial to the interactivity of the system. If a cell is dry in the simulation, it has no meaningful data associated with it. Storing and transmitting these empty values for large portions of the simulation grid would be a waste of memory and bandwidth. We therefore use a sparse representation of simulation data internally, where data values are only stored for wet cells. For discussing this representation, we use an example illustrated in Figure 4.4. The same example is used again in the next section to explain the adaptive surface reconstruction.

In our system, an adaptive grid is a collection of arrays interpreted as several sparsely populated regular grids of different resolutions. These grids, referred to as *levels*, are connected through explicit hierarchy information. Figure 4.4 illustrates an adaptive grid with three levels. Each level has the same world-space extents, but the number of cells in both dimensions doubles for each finer level, while the cell size halves. A typical adaptive grid in our application has six levels and a minimum cell size of one to five meters, depending on the required accuracy of the simulation. Levels are numbered consecutively from the coarsest level to the finest, starting with 0.

Each level-0 cell can be interpreted as the root node of a quadtree which consists of further nodes on higher levels. All *nodes* are uniquely identified by a *node index*. In Figure 4.4, nodes and node indices are shown in green. Each node is also identified by a *grid index* of the form *(x, y, level)*. For example, node 14 in Figure 4.4d has the coordinates (6, 3) counting from the bottom left within level 2 and therefore has the grid index (6, 3, 2).

Hierarchy information assigns either zero or four consecutive child node indices of the next (higher) level to each node by means of the child indices array. If a node has no children, it is a leaf node, referred to as a *cell* of the adaptive grid. Only cells can hold data values (shown in blue in Figure 4.4). Cells are uniquely identified by a *cell index*, starting from level 0 and going from bottom left to top right in each level. Cell indices in Figure 4.4a are shown in red.

To retrieve the data value of the cell corresponding to a given position in the domain, a traversal of the adaptive grid is performed. We demonstrate this process for position p marked in black in Figure 4.4a. As each level-0 node covers its own quadtree, we first determine the level-0 node that covers p. In Figure 4.4b, it can be seen that p is covered by the node with the two-dimensional grid coordinates $(x, y) = (1, 0)$. As the node indices of level 0 are consecutive, the index of this node is $y \cdot w_0 + x = 1$, where $w_0 = 2$ denotes the number of cells of level 0 in x-direction.

77

The children of node 1 can be looked up in the child indices array at index 1, which is given at the bottom of Figure 4.4. In our case, the child indices point to the nodes 4, 5, 6, and 7. The order of child indices corresponds to the four quadrants of the parent node in the order bottom left, bottom right, top left, and top right. From the position of p relative to the node's center position, it is determined in which quadrant to continue with the traversal. In our example, the traversal continues in the top right quadrant. This quadrant corresponds to child node 7 with the grid index $(3, 1, 1)$. We look up the child nodes of node 7, which are 12, 13, 14, and 15. From the position of p within node 7, we decide to continue traversal in the top left quadrant, corresponding to node 14. According to the child indices array, node 14 does not have any child nodes, meaning that it is a cell and the traversal can stop.

As data are only assigned per cell, we need to translate the node index into a cell index by accessing the node-to-cell array at index 14 to retrieve the cell index 11. This concludes the adaptive grid traversal, computing the index of the cell that covers a provided world-space position p. With this cell index, we can finally access the per-cell data array and retrieve the data value 0.3 in our example.

In the important case of water height fields, the adaptive grid is usually only sparsely populated with valid data. Large regions of the domain are not inundated, meaning that the cells' water depth is zero. It would be a waste of memory and bandwidth to always copy, transmit, and store these values. This is why for water height fields, we additionally maintain per-cell validity information in a bit field to indicate which cells have valid data to be fetched. In Figure 4.4a, the invalid cells 0, 2, 3, and 9 are highlighted in red. In order to process only data of valid cells, we introduce the cell-to-data array. Using this array, cell index 11 in our example is translated to data index 7. The data index is then used to look up the data value 0.3 in the compact data array, which is significantly smaller than the per-cell data array. However, this comes at the cost of additional storage for the cell-to-data array, which—assuming 4-byte data types—is just as large as the per-cell data array. The benefit in our case lies in having various data arrays such as relative water depths, absolute water levels, and 2D velocities, which all use the same data indices and therefore can share the cell-to-data array.

As the flood simulation progresses, it is very likely that new cells get inundated, causing their validity to change along with the data values. In this case, the size of the compact data array and the contents of the cell-to-data array will change as well. These are frequent changes likely to occur at each simulation step, so it is particularly important that they can be performed efficiently. The other arrays are mostly static at run time, unless the adaptive grid of the simulation domain itself is changed through user interaction, for example by introducing a water barrier. However, it is important to also reduce the memory footprint of these arrays, as their size becomes large for grids with several million cells. In particular, the array of child indices can be greatly simplified as compared to the explanatory illustration in Figure 4.4. As all four child node indices of a node are always consecutive, it suffices to store only the first one in the array. Also, nodes of the highest-level grid are always cells and never have child indices. Thus, it suffices to

store and maintain child indices for nodes up to the second-highest level. In real-world scenarios, these two strategies shrink the array to around 8 % of its original size.

In the case of a regular grid, which is treated as an adaptive grid with only one level, all arrays needed for traversal are omitted, since the level-0 node indices are the same as the cell indices. Therefore, we only maintain the validity and per-cell data for the terrain, and the cell-to-data indices and compact data for water height fields. This makes dynamic updates much faster and also leads to constant-time data lookups, since no traversal is needed. In our system, we therefore prefer regular grids for smaller scenarios and automatically switch to adaptive grids only if the number of cells exceeds a predefined break-even point, which is approximately eight million cells.

4.4 Adaptive Height Field Reconstruction

The efficiency of interpolation on a regular grid stems from the trivial neighborhood relationship between cells. In a regular grid, the neighborhood of an interior cell is always symmetric and consists of eight neighbors. Data values of neighbors can be accessed easily, which makes operations defined on a larger neighborhood of cells simple. The challenge of interpolation on an adaptive grid is that neighborhood relationships between cells are not straightforward anymore. As the size of neighboring cells can differ by one level, there exists a large variety of arrangements with six to twelve direct neighbors of a cell. This prevents constant-time access to the data of neighbors and makes operations defined on cell neighborhoods—such as interpolation—complex tasks. However, as discussed in the previous section, adaptive grids can be interpreted as a set of sparse regular grids. On each of these conceptual regular grids, we can perform neighborhood operations efficiently again. This interpretation transforms the initially complex operation into multiple simple operations performed on the individual levels. Yet, it also introduces two new challenges, namely, how to reconstruct missing values on the individual levels, and how to combine the results of the individual operations to a final result. Kim and Tsiotras [KT09] perform bilinear interpolation on a local 2×2 grid with missing values reconstructed on the fly. We extend this approach to the 4×4 neighborhood required for bicubic interpolation, in which case the reconstruction of missing values is much more expensive. This is why we represent the individual levels with an actual sparse data structure to cache reconstructed values. As we aim for C^1-continuity of our reconstruction scheme, we also need to take care of level transitions, which has not been addressed in previous work.

In this section, we present a technique for the fast adaptive interpolation or approximation of cell-registered height fields. The main idea is to perform regular interpolation independently for neighboring levels and then stitch the regular regions together with blending functions. We obtain missing values on the individual levels by extrapolation, downsampling, and upsampling of existing values. The resulting reconstruction is identical to bilinear or bicubic interpolation in regular regions and continuous over level transitions. The described technique builds on well-known interpolation functions for regular grids that are interchangeable and independent from the additional challenges of
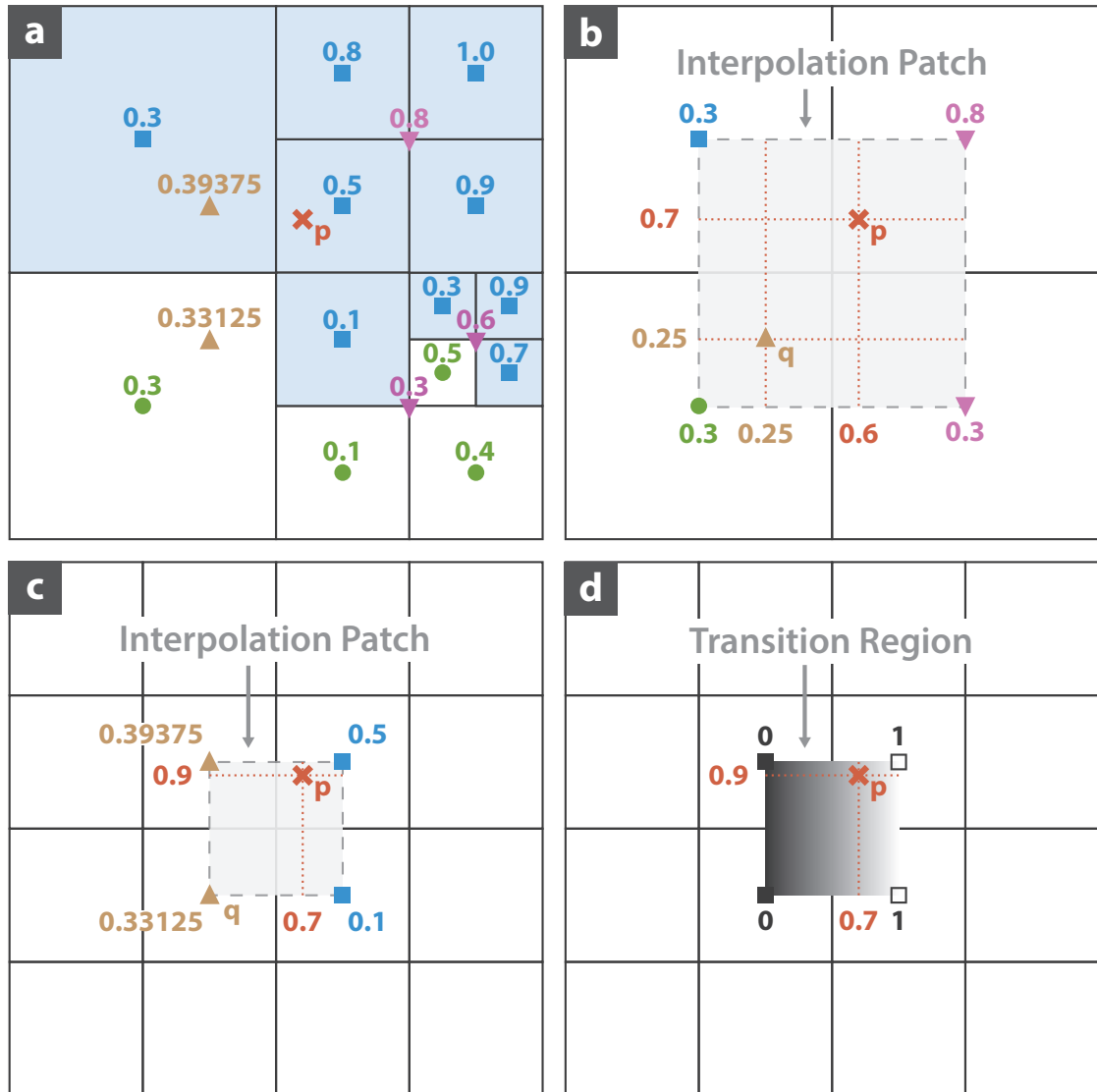
Figure 4.5: Bilinear interpolation within an adaptive grid. (a) All data values needed to calculate an interpolated value at position p on two consecutive levels. Height field values (blue) defined on wet cells are shaded in blue. Dry cells are not shaded. (b) and (c) Missing values have to be reconstructed for the interpolation patches on the two levels. (d) Blending of individual interpolation results within the transition region across levels.

adaptive grids that we tackle. For the sake of simplicity, we therefore start by explaining only bilinear interpolation in detail to highlight the important steps. The subsequent extension to bicubic interpolation then comes naturally. An interactive WebGL application demonstrating both techniques is available online with full source code [Sha].

4.4.1   Bilinear Interpolation

Even in the case of regular grids, interpolation always introduces artifacts in the visualization. In the simplest case of nearest-neighbor interpolation, a constant value is assumed for the entire cell. This, however, results in rectangular structures and discontinuities at cell borders, which strongly contradict our expectations of realistic water propagation.

Bilinear interpolation operates on the 2×2 block of cells surrounding each interpolation position p. We call these cells the *neighborhood* of p in the following. Their centers span a square called an *interpolation patch* over which data values are interpolated. In an adaptive grid, however, the surrounding cells of p do not necessarily have the same size, but might belong to different levels. Yet, two neighboring cells can never differ by more than one level, which is why there are only two cases to distinguish. Either all four cells belong to the same level, in which case we perform usual bilinear interpolation on a locally regular grid, or the four cells belong to two consecutive levels. We now explain how to proceed in the second case with an example illustrated in Figure 4.5. This is the same adaptive grid as already illustrated in Figure 4.4.

The sampling position p, marked in red in Figure 4.5a, is covered by a level-1 cell. While its bottom neighbor belongs to the same level, the bottom left and left neighbors belong to level 0. On each of the two involved levels, we can imagine a 2×2 neighborhood of cells surrounding p, even if not all of the cells exist in the adaptive grid. We call these *imaginary neighborhoods*. The cell centers of each imaginary neighborhood again span a square interpolation patch, as shown for level 0 and level 1 in Figure 4.5b and Figure 4.5c, respectively. The values at the corners of the interpolation patches are the values given at the centers of the corresponding cells of the imaginary neighborhood. Within each interpolation patch, we can perform usual bilinear interpolation of the corner values. However, some of the cells of both imaginary neighborhoods have missing values. Before interpolation, these missing values have to be reconstructed by *extrapolation*, *downsampling*, or *upsampling*.

Extrapolation   On level 0 in our example, only the bottom left and top left cells correspond to leaf cells of the adaptive grid shown in Figure 4.5a. Of these two cells, only the top left cell is a wet cell that has a data value provided by the simulation, which is indicated by a blue square and a blue shade of the cell. The bottom left cell is a dry cell and requires extrapolation, which is indicated by a green circle. To fill in missing values of dry cells, we average over the values of all neighboring wet cells, including diagonal neighborhood. In our application, this extrapolation is only used for absolute water levels to avoid climbing and diving at dry/wet cell boundaries, which is explained in detail in Section 4.5.

Downsampling The top right cell in Figure 4.5b does not exist in the adaptive grid. Instead, this part of the adaptive grid is covered by the four level-1 cells labeled 5, 6, 7, and 8 in Figure 4.4a. Yet, we require a data value at the center of the level-0 cell for the interpolation patch on level 0. We therefore downsample the values of the four level-1 cells by averaging, which is indicated by a purple triangle pointing downward. The bottom right cell on level 0 also needs to be downsampled. In this case, however, not even all corresponding level-1 cells actually exist in the adaptive grid, but only three of them. The value of the last cell itself needs to be reconstructed by downsampling the four level-2 cells 9, 10, 11, and 12 (see Figure 4.4a). It can be seen that downsampling is a recursive process. Furthermore, the cells 2, 3, and 9 covered by the bottom right level-0 cell are dry cells, meaning that extrapolation has to take place before downsampling.

Upsampling With extrapolation and downsampling, all values of the interpolation patch on the coarse level can be reconstructed. On the fine level illustrated in Figure 4.5c, however, we also need to reconstruct missing values by upsampling, indicated by a brown triangle pointing upward. This is necessary for the bottom left and top left cells of the imaginary neighborhood, which are covered by the level-0 cells 0 and 1 (see Figure 4.4a) in the adaptive grid. We focus on the bottom left cell with center q marked in both Figure 4.5b and Figure 4.5c. The value at q has to be reconstructed from the four surrounding level-0 values. These surrounding values have already been gathered for the interpolation patch on level 0, which is why we calculate the missing value by simple bilinear interpolation within this interpolation patch. We perform the same upsampling at the center of the top left cell.

In previous work, Kim and Tsiotras [KT09] performed downsampling and upsampling on the fly. However, during interpolation, reconstructed values are reused multiple times, which is why it makes sense to cache them. In Section 4.4.3, we explain how to store and update all values in a sparse texture hierarchy so that no reconstruction is required during rendering.

Once all required values are present, we perform bilinear interpolation at p on both levels separately. We denote the bilinear interpolation at p on the coarse (level 0) and fine (level 1) levels by $v_c(p)$ and $v_f(p)$, respectively. Since we need a single value, we combine the interpolation results of both levels such that discontinuities at the level transition are avoided. We achieve this with a convex combination of $v_c(p)$ and $v_f(p)$ over a *transition region*, illustrated in Figure 4.5d, which requires both values to be defined over the entire transition region. The transition region is therefore given by the intersection area of the interpolation patches of both levels, which is simply the interpolation patch of the fine level. In this region, the influence $i(p)$ of value $v_f(p)$ should be zero at corners where data values had to be upsampled, and one in the remaining corners. Influence values in between are calculated by bilinear interpolation. The final interpolation value at p is

$$v(p) = (1 - i(p))v_c(p) + i(p)v_f(p). \tag{4.1}$$

In our example given in Figure 4.5, the relative positions of p within the interpolation patches of levels 0 and 1 are (0.6, 0.7) and (0.7, 0.9), respectively. This results in the

individual values $v_c(p) = 0.51$ and $v_f(p) = 0.43825$. With the influence $i(p) = 0.7$, the final result is $v(p) = 0.459775$.

As stated above, our proposed interpolation technique is mostly independent from the interpolation function used for the individual regular levels. It simply blends the individual values together for a continuous surface with smooth transitions between different-sized cells. Bilinear interpolation is a simple and well-known technique ideally suited to introduce the idea. The resulting surface is C^0-continuous and preserves the given data values at cell centers. For many applications, this is a desired and sufficiently realistic representation of the height field data. For water height fields, however, linear surface boundaries still lead to unpleasant visual artifacts such as perfectly straight or angular shorelines (see Figure 4.2b). Furthermore, the C^0-continuity prevents continuous derivatives of the surface, which in the case of water rendering manifests as apparent discontinuities in surface reflections.

4.4.2 Bicubic Reconstruction

For smoother interpolation of data defined on regular grids, cubic splines are a good choice [Ree00]. Compared to the 2×2 cell neighborhood considered for bilinear interpolation, bicubic interpolation takes into account a 4×4 neighborhood of an interpolation position, including not only the four surrounding data points, but also their neighbors. The extended neighborhood allows for matching the derivatives in start and end points of the interpolation splines to neighboring interpolation patches, resulting in a smooth, C^1-continuous surface.

For adaptive cubic interpolation, the increased size of the imaginary neighborhood means that significantly more missing values than for bilinear interpolation have to be reconstructed. For bilinear interpolation, when sampling within a cell, only data of direct neighbors are needed, i.e., reconstruction of existing values on the corresponding level is limited to the 3×3 neighborhood around each cell. Now, we also need data of the direct neighbors' neighbors, which extends the region of values that need to be reconstructed to the 5×5 neighborhood around each cell. First, we need to extend the range of extrapolation. If none of the direct neighbors of a dry cell is a wet cell, we subsequently consider their neighbors as well, and average over the data value of all wet cells in this larger neighborhood. Second, upsampling may be required outside the interpolation patch of the coarse level. For bilinear interpolation, we could conveniently obtain the value of q in Figure 4.5c by bilinear interpolation within the already existing interpolation patch of the coarse level illustrated in Figure 4.5b. For values of the direct neighbors' neighbors, however, we now have to construct separate interpolation patches. This, again, might also require extrapolation and recursive downsampling. In summary, while the individual reconstruction operations are the same as for bilinear interpolation, their combination in the larger neighborhood leads to a high number of required calculations. This makes it even more important to cache the reconstructed values.

Once all values are present, there exist many legitimate options for the cubic reconstruction

filter. Mitchell and Netravali [MN88] define a family of cubic filters with a continuous parameter space ranging from exact interpolation of values by the Catmull-Rom spline to the smoothest approximation by the cubic B-spline. Interpolation preserves the given values at data points, which is often a desired property. Approximation of data values leads to an overall smoother and more natural result than cubic interpolation, as comparable in Figure 4.2. Moreover, approximation with the cubic B-spline can be implemented efficiently with only four instead of the usual 16 texture lookups for the 4×4 neighborhood of an interpolation position by exploiting hardware texture filtering [SH05]. The choice for either interpolation or approximation solely depends on the use case and the desired smoothness. Our reconstruction scheme is completely independent of the used cubic spline.

The final interpolation value $v(p)$ in transition regions is again calculated by a convex combination of $v_c(p)$ and $v_f(p)$. However, to achieve global C^1-continuity, we need at least a cubic blending function $H(t)$, such as the cubic Hermite spline used in the popular *smoothstep* function,

$$H(t) = t^2(3 - 2t). \tag{4.2}$$

The final value is then

$$v(p) = (1 - H(i(p)))v_c(p) + H(i(p))v_f(p). \tag{4.3}$$

The derivative $H'(t)$ of $H(t)$ is continuous, meaning that the convex combination $v(p)$ of the C^1-continuous interpolation polynomials $v_c(p)$ and $v_f(p)$ itself is C^1-continuous within the transition region. Furthermore, $H'(t)$ vanishes on the edges of the transition region, i.e., $H'(0) = H'(1) = 0$. Thus, $v(p)$ is also C^1-continuous on the edges of the transition region, meaning that the height field is continuously differentiable over the whole adaptive grid.

4.4.3 Implementation

Our system uses the adaptive interpolation described above during rendering of the water and terrain height fields multiple times. Both height fields are sampled for triangulation, for surface effects, and coloring, amounting to many million interpolation operations each frame. It is therefore crucial that interpolation can be performed as efficiently as possible using hardware acceleration. This is why we focus on the GPU implementation here, although in our system, interpolation is also used for multiple CPU-side operations such as the water sampling near buildings to determine damage extents.

As described above, we can interpret the adaptive grid as a set of regular grids. For an efficient implementation, we adopt this conceptual representation and maintain a set of regular 2D textures created from the adaptive data structure. We store data values in these textures and update them whenever the input data fields change. For water, we store the absolute water level, the relative water depth, and the two-dimensional velocity for each cell. For the terrain, we store the elevation and an overlay value used for coloring the terrain with a user-defined transfer function. We use multiple texture levels for data

of different grid levels, each having the same dimensions as its corresponding grid level. This is an existing functionality exposed by the OpenGL API. As a consequence, each cell of each individual grid level corresponds to one texel of the multi-level texture. Yet, the use of 2D textures is only partly due to the optimization of texture accesses on GPUs.

More importantly, the individual texture levels also contain the cells of the imaginary neighborhoods that require reconstruction. This enables constant-time access to the cached values after an initial reconstruction step, which is crucial for the real-time applicability of our technique. Of course, allocating video memory for all cells of all levels would not be possible for large data sets, which is the reason for using adaptive grids in the first place. However, as bicubic interpolation within one level requires a 4×4 neighborhood of cells on that level, the set of all possibly needed reconstructed values is limited to the 5×5 neighborhood around each wet cell. With this information, we can greatly reduce the memory requirements by using sparse textures, which are a hardware feature widely supported by modern GPUs. Sparse textures allow for the definition of virtual 2D data fields, where memory is allocated only for manually defined memory pages. We allocate memory only for pages containing at least one cell of the 5×5 neighborhood around any wet cell. As the memory pages have a fixed size (64 KB on our system), the size of the texture's data type controls how many texels one page covers. To minimize memory waste, we minimize the number of texels covered by one page by packing all data fields into different channels of a single multi-channel texture.

We perform the reconstruction and storage of values in the sparse textures whenever input data fields change. We identify the cells that require reconstruction and iterate over these *reconstruction cells* with three different compute shaders in sequence. First, we reconstruct absolute water levels inside dry cells by extrapolation of values from neighboring wet cells and set the relative water depths and velocities of the reconstruction cells to zero. Second, we calculate downsampled values of the four values on the next-higher level covered by each reconstruction cell. As this is a recursive process, we perform this step individually for each level from the finest to the coarsest. Finally, we calculate upsampled values for reconstruction cells covered by lower-level adaptive grid cells.

In summary, we maintain a sparse data structure of all given and intermediate values to separate the reconstruction of missing values from their use for surface reconstruction into two steps. An expensive, but infrequent update step is only necessary when the simulation data change, for example for playback of a scenario or for navigation in time. During highly interactive tasks, such as sketching and manipulation of simulation parameters or navigation in space, we can retrieve the cached values efficiently for fast rendering. For rendering, we create a triangle mesh with view-dependent level of detail on the fly using recursive tessellation, as described in detail in Section 4.6. The elevation of all created vertices is interpolated with our proposed technique. We provide runtime benchmarks for both update and rendering in Section 4.8.

4.5   Artifact Removal

The reconstruction of height field data by interpolation is not always plausible. For example, a reconstructed water surface should have a smooth shoreline that touches the terrain. Furthermore, the water surface should extend to walls that are positioned on dry cells. Here, we assume that the water at dry/wet boundaries near walls should be locally flat. In order to propagate height values to obstacles in a realistic manner, we include boundary conditions for the spline interpolation by properly extending the height field in dry regions. An incorrect extension of height field values leads to unnatural shapes at dry/wet boundaries that suggest false water propagation, which is why we consider them artifacts. At slopes, for example, if missing water levels of dry cells are simply set to the terrain elevation, water climbs uphill (see Figure 4.3a1). In proximity of walls, if the water levels on the wet side of the wall are smoothly interpolated to the lower terrain elevation of dry cells on the other side, the water surface sharply declines towards the wall (see Figure 4.3b1). Here, the water surface might also be interpolated through the wall, leading to ostensible leaking (see Figure 4.3c1).

These artifacts are not specific to adaptive grids or to our proposed interpolation method. Even in the case of traditional bilinear interpolation within a regular grid, these artifacts will occur. The main issue for diving and climbing is an incorrect extension of the water height field to dry grid cells. Leaking is caused by the finite resolution of the simulation domain and the consequent discretization of boundary conditions for simulation, which neglects more precise information. In theory, leaking could be avoided with an ideal interpolator that accounts for dynamically drawn barrier lines, for example for sandbags and dam lines, which can have arbitrarily many vertices. However, even if such an interpolator could be constructed, the computational effort of such high-resolution interpolation would likely be unfeasible for interactive applications. We therefore focus on removing the most prominent artifacts that appear in practice.

Climbing and diving artifacts   Climbing and diving artifacts are different manifestations of the same problem, which is the interpolation of absolute water levels across dry/wet boundaries. In dry cells, no absolute water level is given. However, to interpolate within a 4×4 neighborhood around a given position, values have to be set for these cells. A seemingly good choice is to use the terrain elevation. This is consistent with wet cells, where the absolute water level is the sum of the terrain elevation and the relative water depth, which is zero in dry cells. However, interpolating from the water level to the terrain elevation looks wrong every time we expect the water surface to extend horizontally to a higher obstacle. Climbing occurs if the water height field interpolation connects a higher terrain with the lower water levels. Diving at walls occurs if the interpolation connects the higher water level on the wet side of the wall with the lower terrain elevation on the dry side. Our solution to this problem is already given by the extrapolation of missing values explained in Section 4.4.1. For all dry cells, we extrapolate the absolute water level of all neighboring wet cells and average them. Thus, we extend the geometry of water surfaces by one cell towards dry cells, leading to nearly horizontal intersections

with obstacles (see Figure 4.3a2 and Figure 4.3b2).

Apart from climbing and diving situations, this extension also affects the water surface in shallow regions. This is not desired, because here it gives a false impression of the extents of inundation. We address this issue by a distinction between the surface geometry and the surface visibility during rendering. There, we generate a temporary triangle mesh from the height field, which is rasterized on the pixel grid of the screen into *fragments* by the GPU. We displace the vertices of this triangle mesh using absolute water levels, but decide visibility for each fragment using the relative water depth. For the visibility, we first introduce a threshold for the minimum water depth to display, which will typically be around one millimeter. Such a threshold is generally helpful for rendering water on top of a terrain, because it prevents *z*-fighting. We then calculate the relative water depth of each fragment and compare it to this threshold. If the water depth is less than the threshold, the fragment is discarded, which will happen at the dry/wet boundaries in question. This prevents the depiction of wrong water extents in shallow regions.

A second issue caused by extending the water surface towards the terrain is floating water below the terrain if the water surface geometry intersects with the terrain. In our application, this is unpleasant, because the scene is often viewed from below, for example, for the inspection of sewer networks. Fortunately, we can also fix this issue with the threshold comparison introduced above. As already mentioned in Section 4.4.1, we extrapolate absolute water levels in dry cells, but not the relative water depths, which we set to zero. We can therefore calculate the relative water depth of a fragment in two different ways: Either by interpolating it directly from the simulation data, or by interpolating the absolute water levels and terrain elevations and calculating their difference. We interpret the first way as a ground truth provided by the simulation that leads to visually correct results most of the time, except in climbing situations. In this case, however, the difference between the absolute water level and the terrain elevation will be negative. Thus, calculating the relative water depth in both ways and comparing the smaller of the two values to the threshold allows us to preserve the correct extends of inundation while also removing water below the terrain.

Leaking artifacts   Leaking artifacts as shown in Figure 4.3c1 can also be removed by a per-fragment operation that determines on which side of a wall the fragment is located. These artifacts occur at wall boundaries, which are rasterized on the adaptive grid for the simulation. While the minimum cell size of the adaptive grid is usually one to five meters, a floodwall itself only has a thickness of approximately 15 centimeters. It is therefore not adequately represented by the rasterization, which is illustrated in Figure 4.6a by a white border surrounding the wall cells. Here, wet cells are colored in light blue using nearest-neighbor interpolation to demonstrate that most of the time, the wet cells output by the simulation do not even reach the wall. Bicubic interpolation is therefore needed to extend the water surface to the wall, while at the same time introducing the leaking artifact by extending the surface too much. Our approach to prevent this is to consider the exact geometry of the wall line to determine the location of the fragment

87
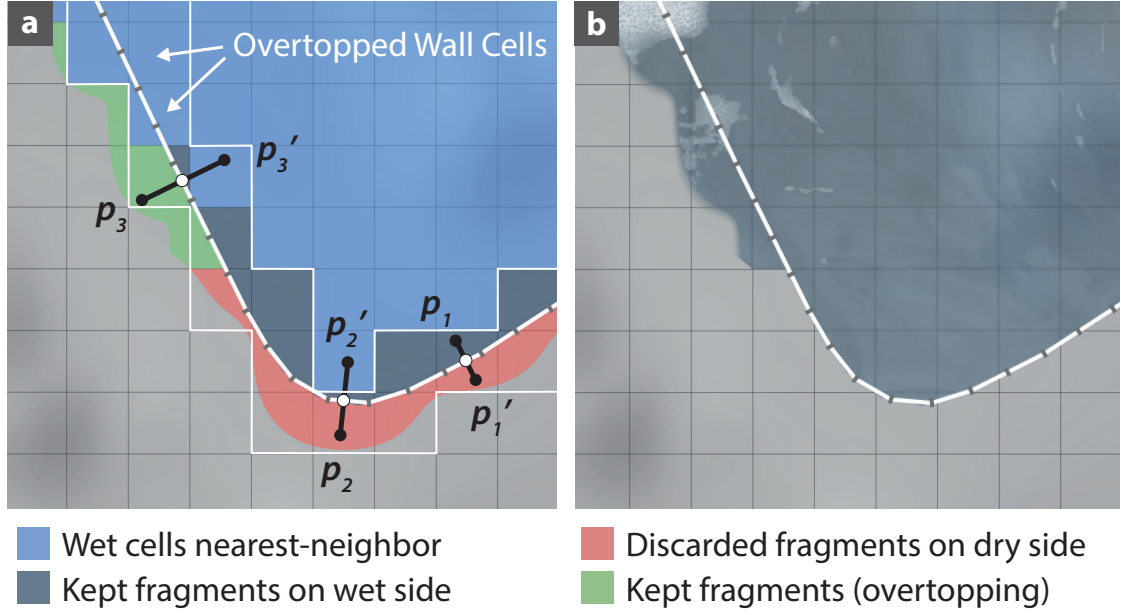
Figure 4.6: Removal of leaking artifacts at walls. (a) Comparison of a wetness value for each fragment p_i and mirror position p_i'. If higher at p_i', p_i is discarded (red), unless the cell containing p_i touches an overtopped wall cell (green). (b) Result without leaking.

relative to the wall, therefore introducing high-resolution data that was not available to the simulation and grid-based interpolation. We explain this approach with the help of Figure 4.6a. For now, we assume that the line geometry is already accessible from within the fragment shader.

We consider three different fragments at positions p_1, p_2, and p_3 contained in dry cells. Starting with p_1, we need to decide whether it is on the flooded side of the wall. We accomplish this by comparing an interpolated wetness value (dry $= 0$, wet $= 1$) at p_1 with the analogous value at the mirror position p_1' on the other side of the wall. We obtain p_1' by reflecting p_1 through its closest point on the wall line. At both p_1 and p_1', we use our proposed interpolation to smoothly interpolate the binary information whether a cell is wet (1) or not (0), which corresponds to the cell's validity introduced in Section 4.3. Unless the wall is flooded from both sides or has been overtopped, this comparison value is significantly higher on the flooded side of the wall. For p_1, we keep the corresponding fragment. For p_2, its mirror position p_2' is closer to a wet cell. Therefore, the interpolated wetness value is higher at p_2' and the fragment at p_2 has to be discarded. However, if the wall is flooded from both sides or has been overtopped, the water surface should be visible on both sides of the wall. To test for this case, we check if the cell containing the fragment in question has a wet neighbor that is either a wall boundary cell or is on the same side of the wall. If this is the case (see p_3), the fragment must not be discarded.

To calculate the point closest to a fragment on any wall line, we need access to the line

geometry in the fragment shader. We rasterize all wall lines on the adaptive grid in a pre-processing step. For each cell, we collect the indices of all line segments of all lines that intersect its 8-neighborhood. The sequences of segment indices of all cells are stored contiguously in a buffer. For each cell, the length and start offset of the sequence is stored. 2D vertex positions of all segments are stored contiguously in a buffer. During rendering, we iterate over all segment indices of the corresponding cell collected before. This is relatively inexpensive, as the vast majority of cells has no line segments in the proximity at all, while most of the remaining cells have less than five. The test also terminates early if the distance to the closest point on any line is greater than twice the square root of the minimum cell size (meaning the maximum distance of influence of a single cell using bicubic interpolation).

4.6 View-Dependent Tessellation

In Section 4.4.3, we described the layout of the height field data in video memory. We now explain how these data are rendered efficiently with view-dependent triangulation using hardware tessellation. As geometry processing on GPUs mostly operates on triangles, a triangulation of the discrete height field data is needed before rendering. However, some of the real-world use cases for flood modeling presented in Section 4.8 use simulation grids with tens of millions of cells, in which case the static triangulation of each individual cell is not feasible on current GPUs. Furthermore, typical workflows in flood management require frequent changes in perspective and zoom level, for example for the setup of large-scale river flooding and then inspection of manually placed protection measures. To support this workflow, rendering has to provide seamless zooming from an overview perspective showing vast stretches of land down to the close-up of individual one-meter cells at interactive framerates. This can be achieved with a view-dependent level of detail incorporated in the triangulation step, for which there exist several established algorithms [PG07]. However, these algorithms date back to a time before widespread support for hardware tessellation and took a lot of effort to compute manually what can now be handled by the hardware. We decided to use hardware tessellation [TBB09] over software approaches for its high performance and simplicity, which makes it immediately applicable.

Rendering with hardware tessellation usually starts by providing a coarse triangle mesh to the GPU's tessellator along with tessellation factors controlling the number of subdivisions. The tessellation factors along the triangle edges are chosen by a metric such that all triangles have approximately the same size in screen space. This automatically leads to a continuous level of detail along the view direction of the camera. Vertices of the resulting triangles can then be displaced according to a height field and are rendered as usual.

However, in our data, the number of cells is already too high to be used as an input for tessellation. Even triangulating only level-0 cells would result in millions of triangles. We therefore resort to a very coarse regular subdivision of the adaptive grid into *chunks* that we triangulate on the fly. This subdivision is based on the dimensions (w, h) of level 0.

In an iterative process, we find the smallest integer divisor d of the grid dimensions such that $\frac{w}{d} \cdot \frac{h}{d} \leq$ 2048. Thus, we subdivide the area covered by the adaptive grid into $\lceil \frac{w}{d} \rceil \times \lceil \frac{h}{d} \rceil$ chunks, resulting in approximately 2048 chunks, which we found to be a good initial subdivision empirically. Whenever height field data change, we compute a 3D bounding box of each chunk from the cells it covers. For each chunk in the view frustum, we generate eight triangles by splitting the chunk along the horizontal and vertical center lines as well as both diagonals. The triangles are stored in a buffer and are used as input patches for hardware tessellation. This largely decouples the rendering performance from the size of the adaptive grid.

A more challenging problem with large data is that the tessellation factor is limited to 64 on current hardware, meaning that the tessellator cannot produce unlimited detail. To achieve a sufficiently fine tessellation of each chunk, much higher tessellation factors than possible would be required. We solve this issue by recursive tessellation of triangles in multiple passes, similar to the approach proposed by Lee et al. [LJL13]. In the first pass, the input triangles for tessellation are the triangulated chunks. Tessellation factors for the triangle edges are calculated such that no resulting triangle edge would be longer than five pixels when rendered. For vertex spacing, we choose the built-in *fractional_odd_spacing* behavior, which leads to smooth transitions between tessellation factors. If the calculated levels exceed hardware limits, we clamp them to the allowed maximum and set a flag that indicates that this triangle is not sufficiently subdivided. In the geometry shader, we evaluate this flag for each triangle. If it is false, the triangle is rasterized and drawn. Otherwise, its vertices are written to a buffer with the help of an atomic counter. This buffer has the same layout as the triangle vertex buffer containing the chunk triangles, which is why they are interchangeable. Once all triangles are processed, the previous output buffer is swapped with the input buffer and tessellation is invoked again for the new input triangles. This continues until all triangles are smaller than the predefined size in pixels. We terminate tessellation after at most three passes, although in practice, a third pass is rarely ever needed. It is difficult to estimate how many triangles have to be stored in a buffer for further subdivision. For the allocation of the buffer objects, we assume two million triangles. If more triangles need to be stored during geometry shader execution, we still increment the atomic counter, but stop writing to the full buffer. Once the pass is finished, we resize the output buffer to the required number of triangles and restart the pass with the previous input, but do not render already rendered triangles again. With this resizing behavior, we avoid overly pessimistic memory provisioning at the cost of slightly longer rendering durations of a few frames at the beginning of the application.

4.7 Water Flow Perception

If the simulation domain is slowly flooded, it is very obvious where the water is flowing. Once the terrain is inundated, however, flow behavior such as direction and speed are no longer visible from the water surface itself. Common visualization techniques such as

glyphs convey this information in a very expressive way, but add to the complexity of the visualization. This is a concern, for example, if visualizations of simulation results need to be communicated to the general public. While the appearance and behavior of water are well-known and are intuitively recognized [KC14], more abstract visualization techniques are not. In situations where only subtle hints should be provided, animated waves and foam can be used as intuitive and less obtrusive alternatives [Vla10, GSH+15]. In contrast to previous work, we do not focus primarily on physically plausible water rendering. Instead, we use waves and foam as visual metaphors for flow directions and high velocities. The size of waves and the intensity of foam are deliberately exaggerated such that they are also visible from an overview perspective. Also, flooding scenarios usually span several hours or days, but are replayed in time lapse. Waves moving at physically plausible speeds as given by the simulation would not be perceivable at all in this application. As waves and foam dynamically change with the simulation data over time, we refer to the video accompanying the original paper for a demonstration [Vid].

For open water such as oceans, very realistic waves can be created by the superposition of individual wave functions with different wavelengths and amplitudes [Tes01, Fin04, NSB13, Hor15]. The parameters are usually defined manually by artists or are controlled by probabilistic models. The large variety of the summed waves leads to a complex function with both small-scale and large-scale features that is a visually plausible approximation of real waves. However, these waves are largely unbounded and defined globally, making it hard to include local behavior such as varying flow directions, turbulence, and interaction with floating objects and the shore. For local flow behavior, this approach has to be combined with local approaches such as particle simulations, which are only feasible in real time at a very coarse resolution or within a very small area.

Tile-based wave synthesis [Vla10, Gri11, vH11, GH12], on the other hand, is based on texture animation [MB96, Ney03] and is ideally suited for local flow behavior. The spatial domain is discretized into uniform tiles, to which a velocity is assigned according to a flow or velocity field. Within each tile, a pre-computed, toroidal wave pattern in the form of a displacement map or normal map is aligned towards the flow direction. This causes discontinuities at tile borders, which can be hidden by using overlapping tiles, such that a smooth transition between the waves of neighboring tiles can be achieved by blending. For animation, the texture is offset along the flow direction within each tile over time. While this approach enables the visualization of flow features as small as the tile resolution, it also limits the maximum wavelength that can be included in the pre-computed wave pattern. This makes previous tile-based approaches unsuitable to visualize large-scale principal flow directions. This shortcoming is amplified by the use of a single wave pattern for all tiles, leading to very apparent repetitions of wave arrangements over larger water surfaces. Furthermore, if the underlying velocity field is changing over time, the texture offset needs to be reset periodically to avoid distortions of the wave pattern. This reset has to be hidden as well by blending over time, which itself leads to a distinctive pulsing of the waves.

We aim to combine the strengths of both approaches by integrating wave superposition
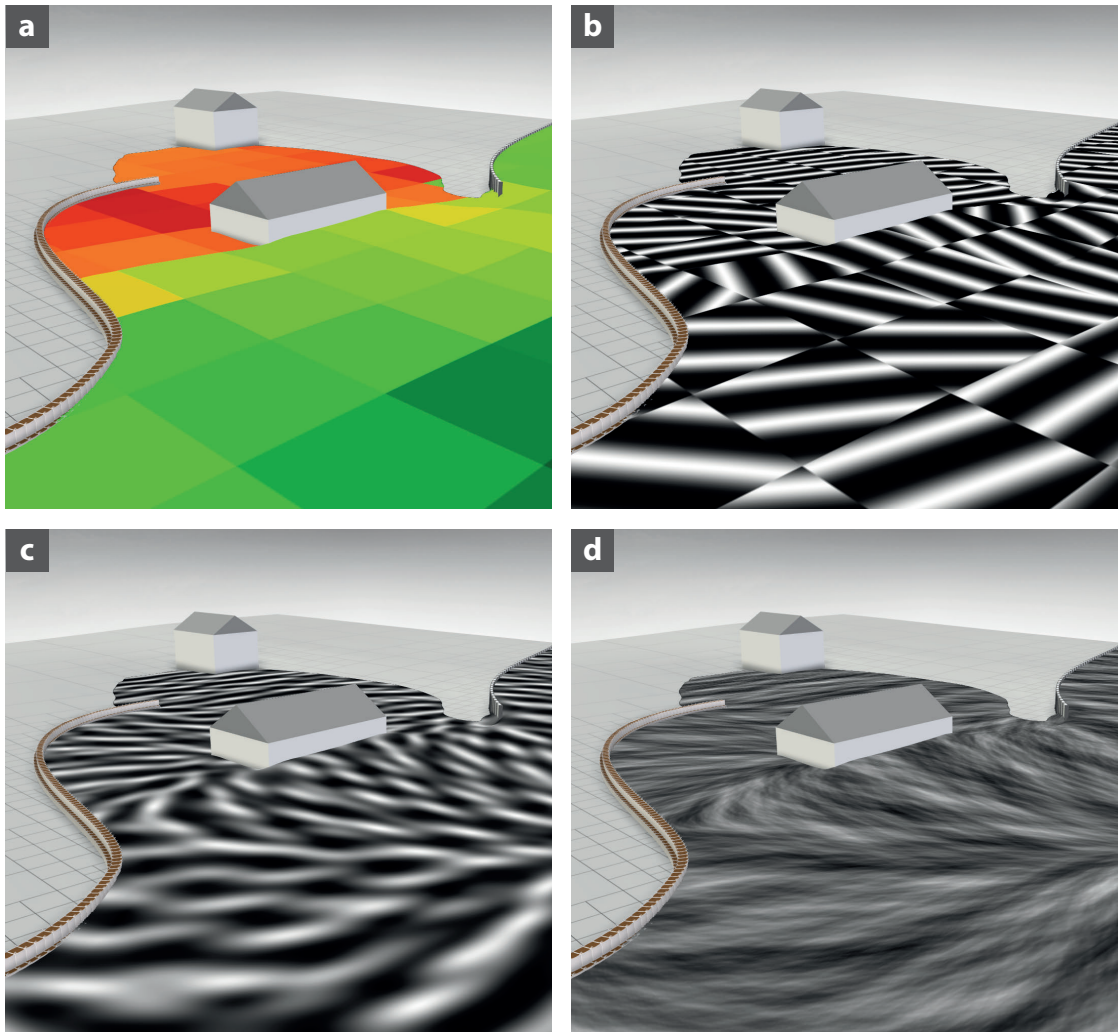
Figure 4.7: Overview of tile-based wave synthesis. (a) Average velocity of each tile for a single wavelength, normalized for display. (b) Sine waves of a single wavelength oriented towards the flow direction. (c) Blended waves of a single wavelength. (d) Superposition of sine waves of multiple wavelengths for vertex displacement.

into tile-based wave synthesis to replace the pre-computed wave patterns. By using continuous wave functions rather than static textures with wave patterns, we avoid repetitive wave arrangements and eliminate pulsing artifacts. We use tiles of different sizes for waves of different wavelengths, which allows us to visualize small-scale local flows on top of large-scale principal flows with arbitrary wavelengths. The individual steps of our approach are illustrated in Figure 4.7. For multiple wavelengths, the spatial domain is subdivided into tiles over which an average velocity is calculated (see Figure 4.7a). According to the velocity, sine waves are oriented inside the tile (see Figure 4.7b), which leads to discontinuities at tile borders. This process is performed for four partially
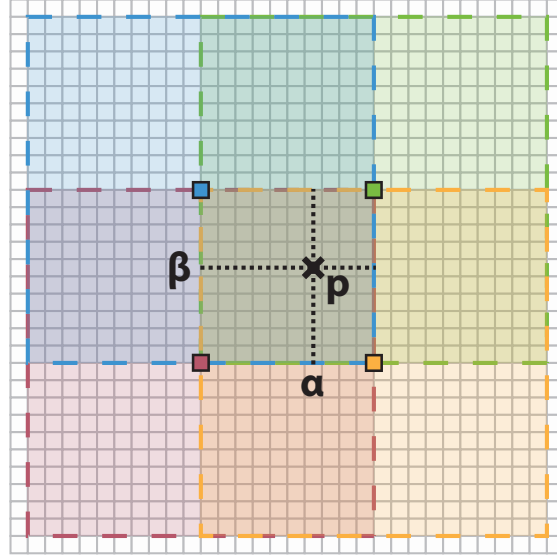
Figure 4.8: Layout of four overlapping tiles around sample position p for seamless blending of values between tiles.

overlapping tiles (see Figure 4.8) to get four different wave heights at each position p. The final wave height is interpolated between these four wave heights, which eliminates the discontinuities (see Figure 4.7c). The wave heights of all considered wavelengths are summed up to a final displacement value (see Figure 4.7d) to be applied to triangle vertices of the water surface.

Within the tiles of different sizes corresponding to different wavelengths, we generate individual waves by a wave function that extends the simple sine function proposed by Finch [Fin04]. The extensions allow us to control the chaotic behavior and adjust the amplitude of waves according to simulation data. For n different wavelengths $\lambda_i = r_i 2^i \lambda$, we define a tile over which the wave function is evaluated. λ is the shortest considered wavelength and $r_i \in [0.8, 1.0]$ is a random value preventing exact frequency doubling. We chose $n = 5$ and $\lambda = 0.5$ m empirically. Given a world-space position $p = (x, y)$ at time step t, we define a tile $T = [x_0, x_1] \times [y_0, y_1]$:

$$(x_0, y_0) = 2\lambda_i \left\lfloor \frac{(x, y)}{2\lambda_i} \right\rfloor, \quad (x_1, y_1) = (x_0, y_0) + (2\lambda_i, 2\lambda_i). \tag{4.4}$$

This tile is shaded in blue in the upper left of Figure 4.8. Additionally to this tile, we consider the green tile shifted horizontally by λ_i, the red tile shifted vertically by λ_i, and the orange tile shifted both horizontally and vertically by λ_i, for which we evaluate the wave function separately. Over each of these tiles, we require an averaged velocity $v_{i,T}$ as well as the average flow direction $d_i = v_{i,T} / \|v_{i,T}\|$. The average velocity $v_{i,T}$ in the area covered by the tile T is computed by integrating the velocity field over the tile. For a fast approximation, we access the velocity field texture introduced in Section 4.4.3

with hardware-supported linear filtering between the two texture levels with cell sizes closest to the current wavelength. For regular grids, the accuracy of integration can be increased with little effort by creating a summed-area table (SAT) of the velocity field texture using a parallel prefix-sum algorithm [SWH15] in a preceding compute shader. Each texel of the SAT texture contains the sum of all velocities to the left and below the texel. From this representation, the average velocity $v_{i,T}$ of the current tile can be calculated in constant time as

$$v_{i,T} = \frac{SAT(x_1, y_1) - SAT(x_0, y_1) - SAT(x_1, y_0) + SAT(x_0, y_0)}{n_T}, \qquad (4.5)$$

where n_T is the number of texels that T covers in the velocity field, i.e., the tile size divided by the cell size.

In analogy to the notation by Finch [Fin04], we denote the wave height at world-space position $p = (x, y)$ at time step t for n different wavelengths as

$$H(x, y, t) = \sum_{i=0}^{n-1} A_{i,T} W_{i,T}(x, y, t) \qquad (4.6)$$

with the wave amplitude

$$A_{i,T} = \frac{\|v_{i,T}\|}{\|v_{\max}\|} \frac{w_i}{\sum_{j=0}^{n-1} w_j}, \quad w_i = \left(\frac{1}{2} + 1\right)^{-i}, \qquad (4.7)$$

and wave function

$$W_{i,T}(x, y, t) = 2 \left(\frac{\sin(-\theta_i \omega_i + t\varphi_i + r_T 2\pi) + 1}{2}\right)^2 - 1 \qquad (4.8)$$

for each tile T. $\theta_i = \langle d_i, (x, y) \rangle$ is the distance of the wave traveled along the average flow direction and $\omega_i = 2\pi/\lambda_i$ is the wave frequency. $\varphi_i = \min(r_i \omega_i, \pi/2)$ is the phase constant, which controls the traveling speed of the wave. $r_T \in [-0.1, 0.1]$ is a random phase offset of tile T that is used to introduce wave irregularities between neighboring tiles. Amplitude $A_{i,T}$ depends on the average velocity $v_{i,T}$ normalized by a maximum velocity $v_{\max}$, which is either provided by the user or derived from the velocity field. As a result, waves are higher in high-velocity regions, and there are no waves in regions at rest.

Tiling introduces discontinuities at the tile borders, as visible in Figure 4.7b, which need to be treated to avoid artifacts in the visualization. This is why for each (x, y, t), we evaluate the four wave functions $W_{i,T}(x, y, t)$ in the four tiles covering $p = (x, y)$ and blend them such that the weight of each tile smoothly fades from one at its center to zero at its borders. Figure 4.8 illustrates the blending in the overlapping region of the four tiles at p, which is a bilinear interpolation with the horizontal and vertical weights α and β, respectively. The straightforward choice for the blending weights is a linear decrease from the tile center to its borders, i.e.,

$$\alpha = 2 \left|\mathrm{frac}\left(\frac{x}{2\lambda_i} + \frac{1}{2}\right) - \frac{1}{2}\right|, \quad \beta = 2 \left|\mathrm{frac}\left(\frac{y}{2\lambda_i} + \frac{1}{2}\right) - \frac{1}{2}\right|. \qquad (4.9)$$
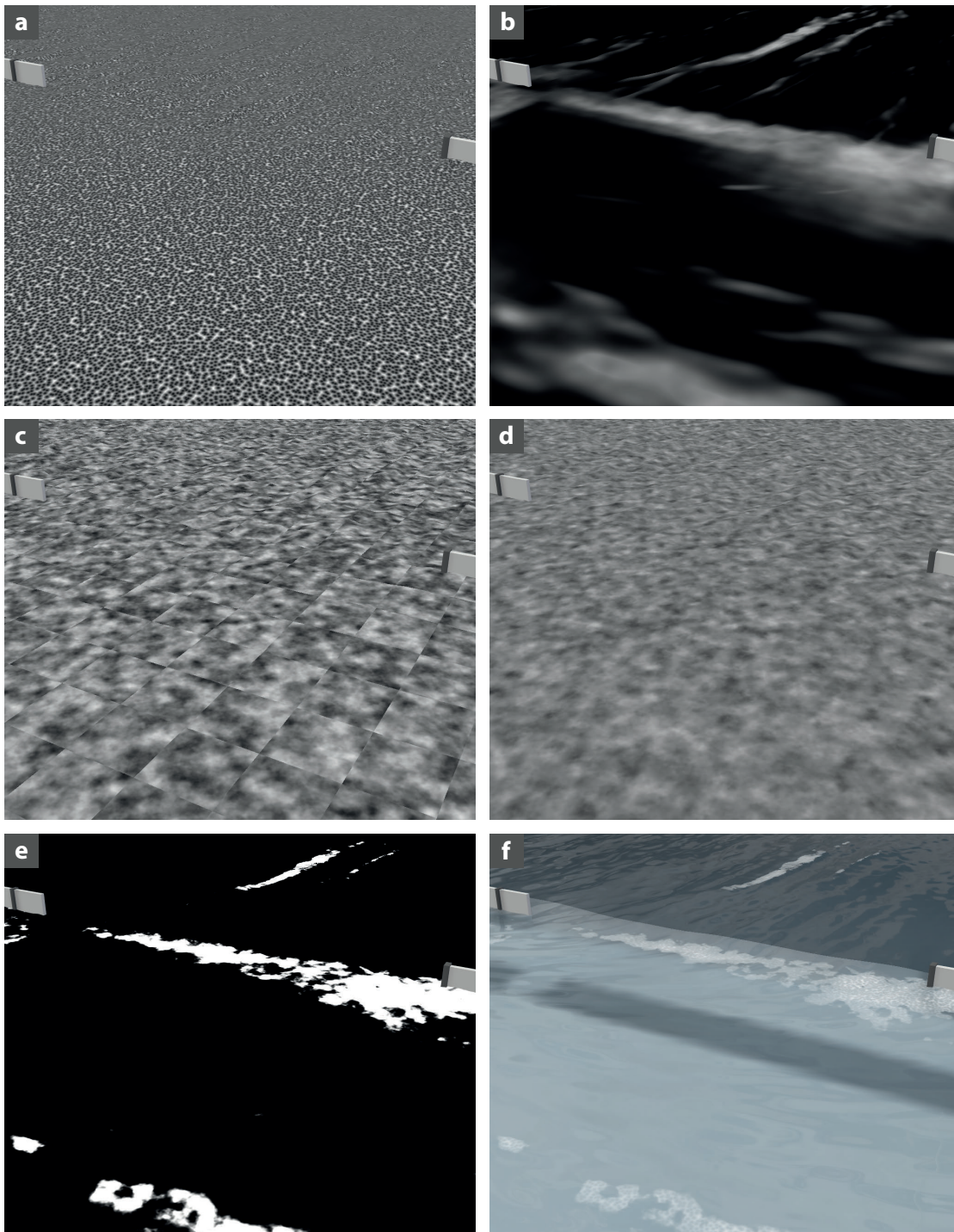
Figure 4.9: Overview of tile-based foam synthesis. (a) Animated cellular noise. (b) Foam weights. (c) Gradient noise per tile used for thresholding. (d) Gradient noise blended between tiles. (e) Foam mask after thresholding. (f) Application of masked cellular noise.

However, our experiments have shown that linear blending weights result in visible blending regions at tile borders. We avoid these artifacts with continuously differentiable weights,

$$\alpha = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{x}{2\lambda_i}2\pi\right), \quad \beta = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{y}{2\lambda_i}2\pi\right). \tag{4.10}$$

The blended result $W_i(x, y, t)$ for a single wavelength is shown in Figure 4.7c. The final wave height $H(x, y, t)$ over all wavelengths used for vertex displacement is shown in Figure 4.7d.

Based on the wave tiles, we also generate foam on the water surface to highlight areas of high velocity and significant changes in the velocity, i.e., high magnitudes in the velocity field's gradient field. Additionally, we add foam along high waves to increase their visibility. We model the foam in regions with high magnitudes as contiguous patches of white bubbles with frayed contours and hard transitions to the water. To achieve this, we first apply an animated Worley noise [Wor96] pattern to the water surface, which is a procedural cellular noise function that gives the foam a bubble-like appearance (see Figure 4.9a). The visibility of this cellular noise function is modulated by a mask depending on a foam weight function derived from the velocity field's gradient field and the previously synthesized waves (see Figure 4.9b). To obtain the foam mask, the foam weight function is thresholded with an animated gradient noise function. The gradient noise is first computed per tile (see Figure 4.9c) and then blended between overlapping tiles to avoid visible seams at tile borders (see Figure 4.9d). The mask resulting from the threshold comparison (see Figure 4.9e) then determines the opacity of the cellular noise function on the water surface (see Figure 4.9f).

For serious applications like the exploration of simulated flooding scenarios in flood management, aesthetic rendering of water surfaces has to be balanced with the utility for specific tasks. For example, a realistic depiction of a flood would be very turbid and opaque, making it hard to estimate water depths from a still image. We therefore reduce the degree of realism to emphasize important features of the water flow, while still offering aesthetic results. We include visual hints of the water depth by using a transfer function [Bel04] for water color and opacity, such that inundated structures remain visible and depth information is preserved. As a result, the water opacity is well below the realistic limits for turbid water, which allows the viewer to spot otherwise invisible inundated objects below the water surface. We also apply a depth-dependent box blur [HSC+05] to refractions such that objects in deep water regions are blurred and distorted more than in shallow regions. The effect of these techniques can be seen in Figure 4.10 in comparison to an actual flood in Cologne, Germany, where the opacity of the water makes it impossible to estimate water depths or identify inundated structures.

4.8   Results

In this section, we demonstrate the results of our approach in various typical use cases for flood management. Based on the working task and the specific scenario, these cases
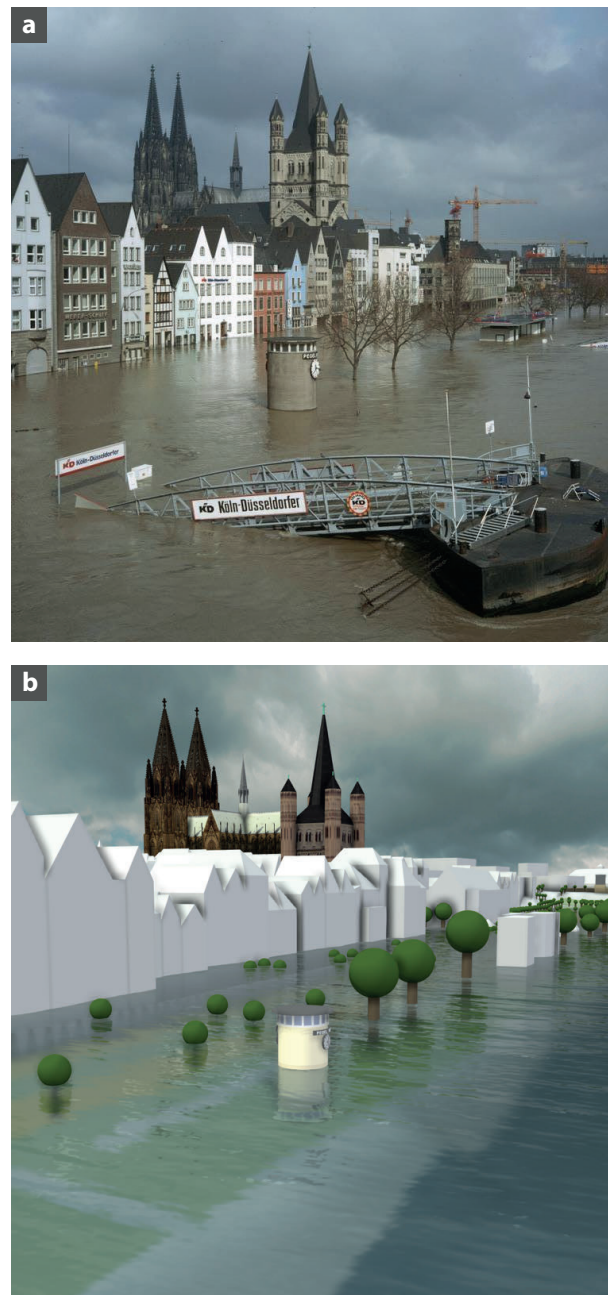
Figure 4.10: Comparison of real flooding with our technique. (a) Turbid water in reality makes it impossible to identify completely inundated structures and to estimate water depths. (b) Our technique provides a geospatial context by revealing inundated structures such as streets and floodwalls.
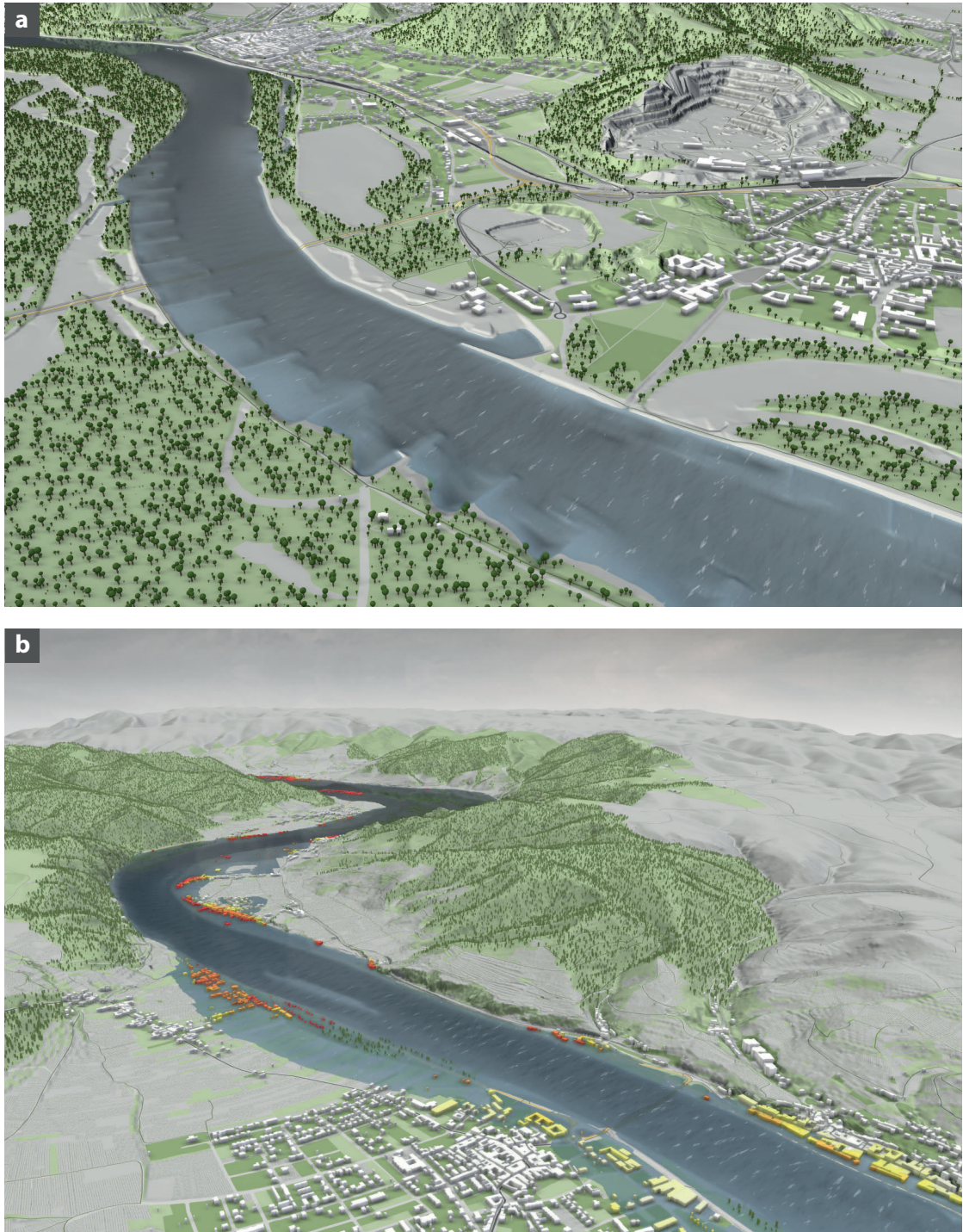
Figure 4.11: (a) Use case **C1**: Large-scale flood risk assessment in the Marchfeld region in Austria. The water shading reveals the structure of the river bed. (b) Use case **C2**: Flood risk assessment in the Wachau region in Austria.
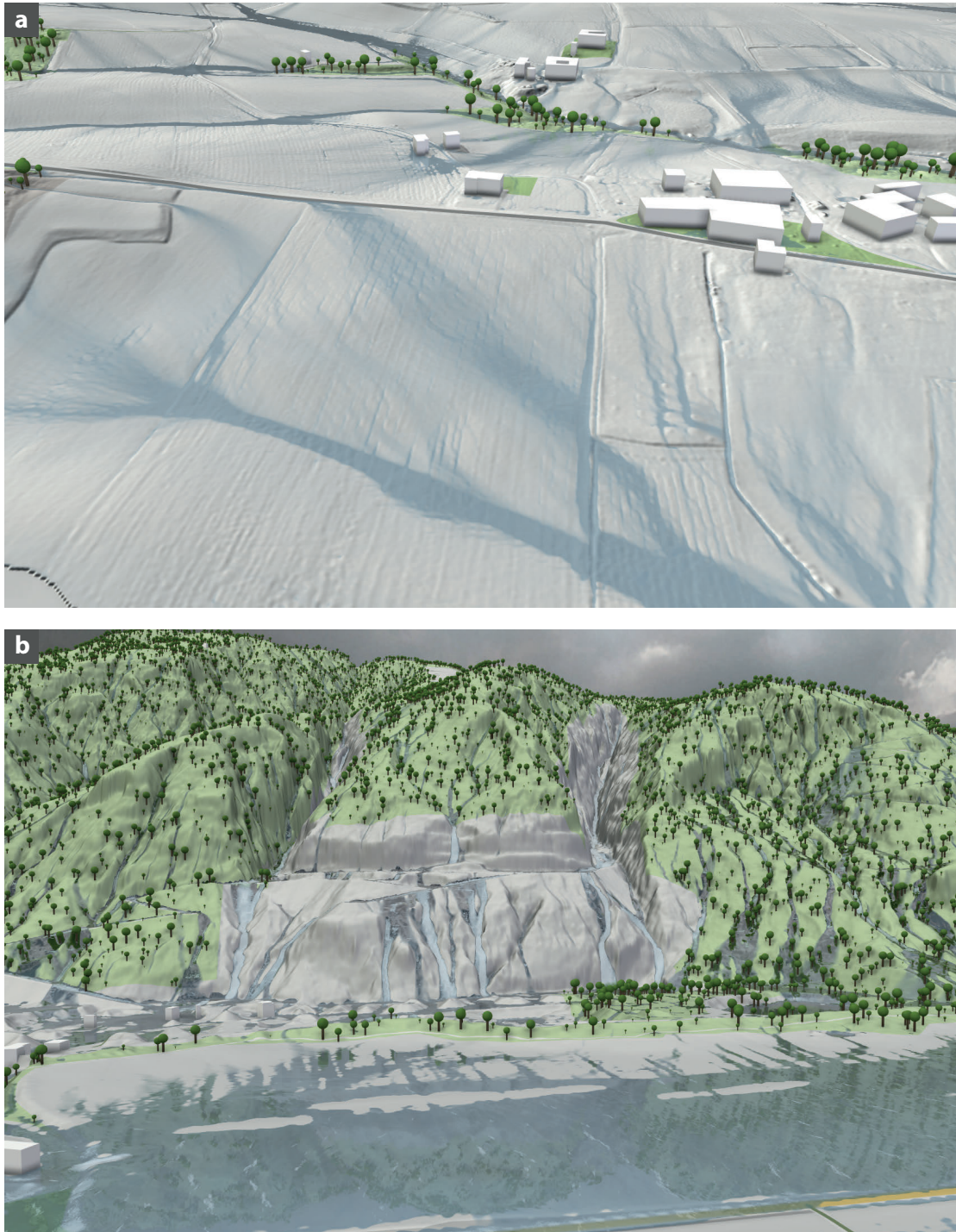
Figure 4.12: (a) Use case **C3**: Stormwater runoff on a high-resolution grid in Petzenkirchen, Austria. (b) Use case **C4**: Stormwater runoff in a mountainous region in Tyrol, Austria.
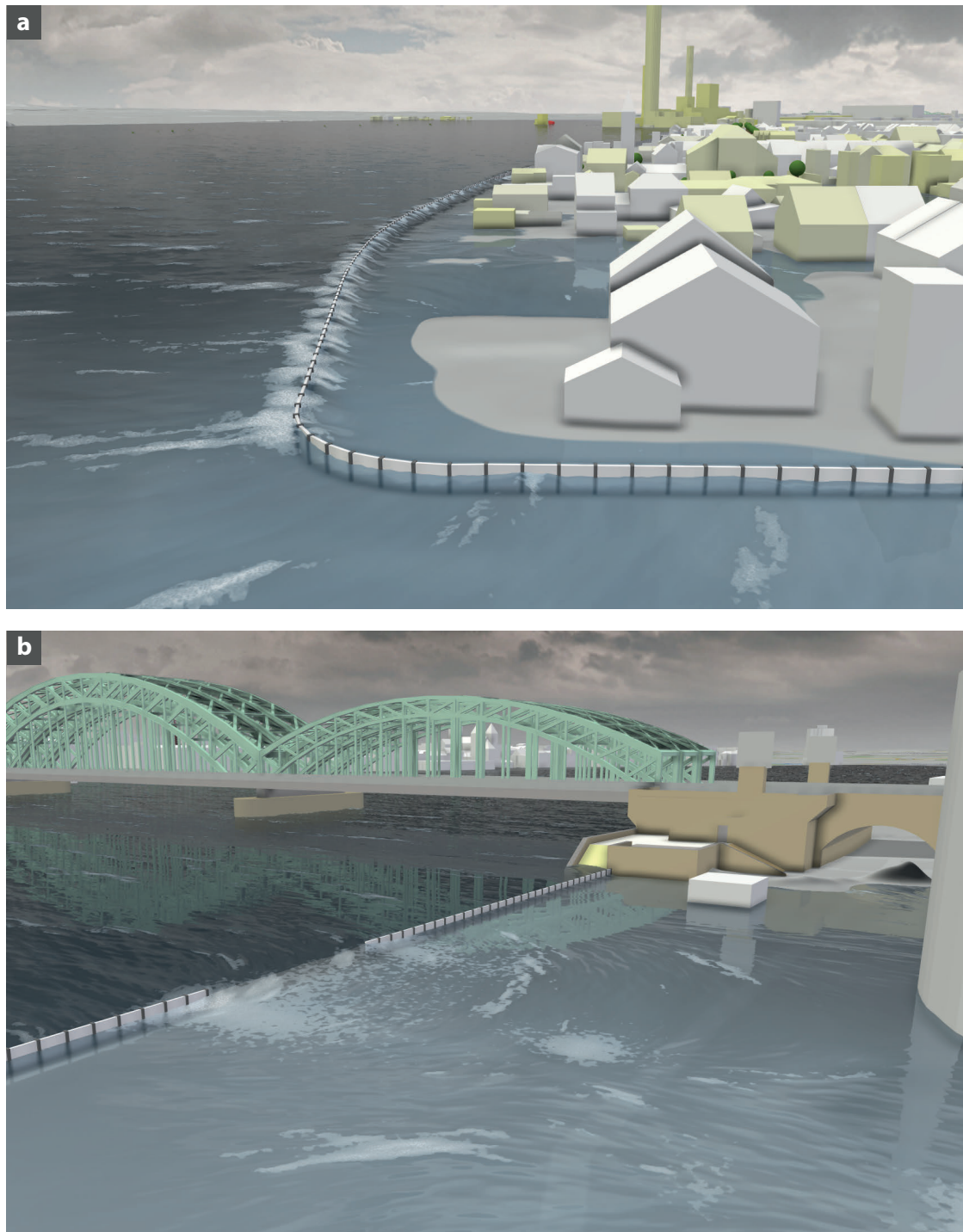
Figure 4.13: Use case **C5**: Failing floodwall scenarios for interactive planning of protection measures in Cologne, Germany. (a) Current overtopping regions are indicated by foam. (b) Directional waves and foam indicate principal flow directions after a breach.
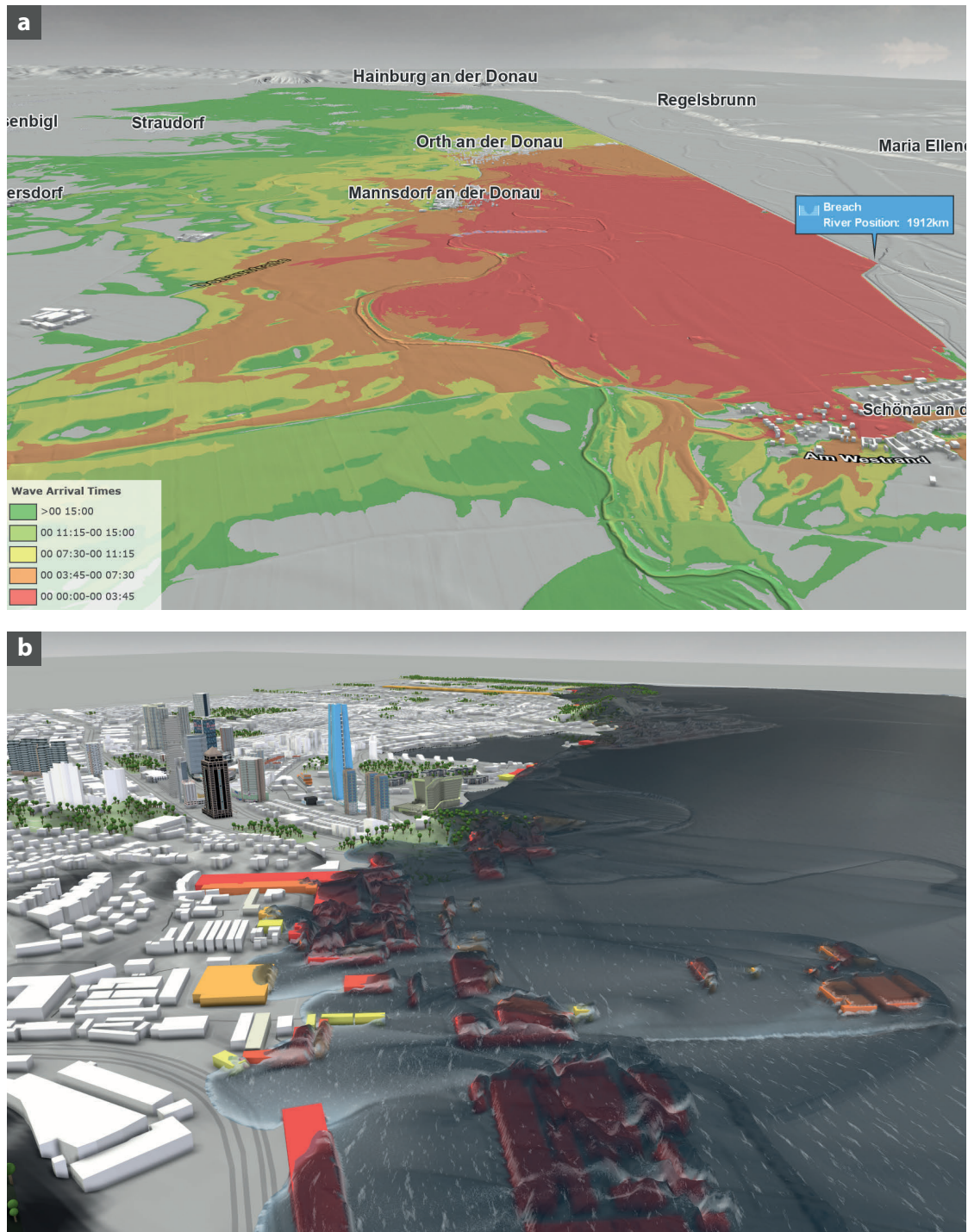
Figure 4.14: (a) Use case **C6**: Terrain overlay of interpolated wave arrival times after a dike breach in the Marchfeld region in Austria. (b) Use case **C7**: Tsunami impact on a city in China.

Use Case	Extents	Cell Sizes	# Cells	# Wet	T. Data	W. Data
C1 (initial)	17.0×19.4 km^2	$3 \ldots 96$ m	22.58 M	0.75 M	1553 MB	1933 MB
C1	"	"	"	5.32 M	"	"
C2	9.8×4.0 km^2	3 m	4.31 M	0.62 M	159 MB	211 MB
C3	2.0×2.1 km^2	0.5 m	17.15 M	15.12 M	625 MB	822 MB
C4	4.2×5.7 km^2	2 m	6.01 M	5.95 M	221 MB	293 MB
C5	4.1×5.1 km^2	5 m	1.00 M	0.34 M	37 MB	57 MB
C6	31.9×13.4 km^2	$3 \ldots 96$ m	12.51 M	3.31 M	1067 MB	1553 MB
C7	9.4×6.6 km^2	3 m	7.29 M	3.55 M	268 MB	355 MB

Table 4.1: Description of case studies. Columns from left to right: Name of the use case, extents of the simulation domain, cell sizes of individual levels, number of cells, number of wet cells, terrain data size, water data size.

differ significantly in terms of simulation grid extents, grid cell sizes, and the share of wet and dry cells. We show exemplary results of various scenarios and provide the scenario parameters in Table 4.1 together with runtime benchmarks in Table 4.2. For more use cases and animated results, we refer to the video accompanying the original paper [Vid]. For benchmarking, we use a system with an Intel Core i7-6700K 4 GHz CPU, 64 GB RAM, and an Nvidia GTX 1080 Ti GPU. The rendering resolution is 1920×1080.

The first use case **C1** involves the modeling of river flooding of an outstanding magnitude such as once-in-a-century floods for flood risk assessment. For this task, engineers need to frequently navigate in space and time and switch between overview and detail perspectives to investigate flood risks on a region, village, or infrastructure level. Figure 4.11a shows the Danube river in the Marchfeld region in Austria, with a simulation domain spanning many villages over an area larger than 300 km^2. For comparison, we provide timings for the initial state and after seven days of flooding. In this case, the large scale of the data makes the use of an adaptive grid essential. As use case **C2** shown in Figure 4.11b, we consider a second, small-scale scenario with a regular grid. When modeling river flooding, only a fraction of cells of the simulation grid is wet, which makes the amount of inundation visible from the extents of the water surface.

In contrast, stormwater and surface runoff modeling operates on high-resolution grids where almost all cells are wet. Use case **C3** is an open-air hydrological laboratory in Petzenkirchen, Austria, shown in Figure 4.12a. Use case **C4** shown in Figure 4.12b is a mountainous region in Tyrol, Austria. Here, it is important to visualize where water collects and forms small streams. Showing all wet surfaces at full opacity, which in the case of rainfall is the entire scene, would hide this information. This is why we reduce the opacity of the rain layer if it is below one millimeter.

The interactive planning of urban protection measures requires smaller regions and coarser grids (3–5 m resolution) to enable on-the-fly simulation. This usually makes the artifacts stemming from water surface reconstruction even more prominent. Our use case **C5** shown in Figure 4.13 is located in the city of Cologne, Germany. Here, a floodwall is

Use Case	Update	Terrain	Water	Artifacts	Flow Visusalization
C1 (initial)	57.8 ms	4.9 ms	3.1 ms	0.4 ms	2.7 ms
C1	68.8 ms	"	9.6 ms	0.7 ms	2.6 ms
C2	7.0 ms	2.1 ms	6.7 ms	0.1 ms	0.4 ms
C3	36.5 ms	1.7 ms	7.7 ms	0.7 ms	2.1 ms
C4	14.5 ms	2.7 ms	3.3 ms	0.7 ms	5.0 ms
C5	2.5 ms	1.7 ms	4.9 ms	0.5 ms	1.9 ms
C6	59.7 ms	5.8 ms	8.1 ms	0.5 ms	1.6 ms
C7	11.8 ms	2.2 ms	5.3 ms	0.3 ms	2.4 ms

Table 4.2: Benchmarks for case studies. Columns from left to right: Name of the use case, water data update time, terrain rendering time, water rendering time with tessellation and surface shading, artifact removal time, visualization time of flow properties with waves and foam.

failing due to overtopping or a breach. In both cases, waves and foam automatically indicate flow directions and high velocities. In such scenarios, data updates and rendering need to be fast to support fully interactive sketching of protection measures, for example.

In use case **C6**, we demonstrate the analysis of breach scenarios of a dike more than 50 km long along the Danube, again in the Marchfeld region. Engineers define different breach locations and their structure along the dike, and then navigate in both time and space to analyze the simulation results, for example to identify damage to villages and important infrastructure, and to check wave arrival times for evacuation planning. Figure 4.14a shows wave arrival times interpolated with our proposed method and visualized as terrain overlay according to a user-defined transfer function. Based on the simulation results, the engineers identify good locations for local protection barriers to protect parts of villages where possible, which can be sketched as lines directly on the terrain. The robustness and construction details of these barriers are then evaluated in further simulation runs.

In our last use case **C7** shown in Figure 4.14b, the impact of a tsunami on a Chinese city is modeled, which results in high water velocities and complex waves. Inundated buildings are colored by their estimated damage according to a user-defined transfer function.

For all use cases, we also assessed the accuracy of our proposed surface reconstruction, provided in Table 4.3. The established procedure for this task is to calculate the root mean square error between the reconstructed surface and a high-resolution ground truth [Ree00, Kid03], for which we use the original digital elevation model of the terrain defined on a regular grid. Even for adaptive height fields, we could reproduce the results of Kidner [Kid03] showing that third-order interpolation results in a more accurate surface reconstruction than linear or nearest-neighbor interpolation. As expected, B-spline approximation has a higher deviation from the ground truth than exact cubic interpolation (Catmull-Rom), as it does not preserve values at data points for the sake of a smoother surface.

Use Case	Nearest Neighbor	Bilinear	Catmull-Rom	Cubic B-spline
C1	0.044 m	0.029 m	0.023 m	0.037 m
C2	0.503 m	0.126 m	0.101 m	0.173 m
C3	0.075 m	0.036 m	0.028 m	0.044 m
C4	1.177 m	0.345 m	0.273 m	0.481 m
C5	0.280 m	0.240 m	0.210 m	0.277 m
C6	0.199 m	0.182 m	0.166 m	0.208 m
C7	0.307 m	0.133 m	0.121 m	0.149 m

Table 4.3: Accuracy of surface reconstruction. The root mean square error between the terrain surface and the original digital elevation model expresses the mean deviation from the ground truth.

4.9 Evaluation

We conduct our research in close collaboration with domain experts working in the field of flood management, hence evaluation of our results is a continuous process. In evaluation sessions prior to this work, shortcomings in the previously used visualization of water surfaces have emerged, which triggered our research for the solutions presented in this chapter. These solutions were evaluated qualitatively by domain experts in live sessions, which focused on the differences to previously used techniques and how they affect the experts' everyday workflow. To strengthen this with quantitative results, we conducted a survey among experts in various fields of work as well as the general public.

4.9.1 Live Sessions

During two separate live sessions of about one hour each, we demonstrated our solutions in different real-world flooding and heavy rain scenarios to four flood management experts from two different organizations. The first expert works for the flood protection center of Cologne, Germany, a leading institution for integrated flood management in Europe [Köl]. Her typical tasks in urban flood and stormwater management include heavy rain modeling, response planning, and public communication. The other three experts are engineers at the consulting agency RIOCOM [RIO] focusing on flood risk management in Austria for over 20 years. Their typical tasks include flood risk analysis, the creation of flood risk maps, and the design of protection measures, for which they use our decision support system Visdom. For evaluation, we showed the experts various prepared flooding scenarios in Visdom and asked them to fulfill specific tasks. The first task was the subjective assessment of the usefulness and quality of our proposed techniques for the use cases discussed in Section 4.8. For the second task, we showed the experts the same scenario with different interpolation methods as well as with and without common interpolation artifacts. We asked them to compare the different visualizations and indicate their preferences. Finally, we asked them to identify local flow directions as well as deep and shallow regions of inundated regions from an overview perspective.

After being shown results of nearest-neighbor, bilinear, and bicubic interpolation as well as B-spline approximation, all four experts concurred that B-spline approximation led to the most aesthetic results and was also the best surface reconstruction to use for their working tasks. The engineers said that the rectangular structures of nearest-neighbor interpolation were unpleasant to look at and carried no valuable information. The stormwater expert liked to have several options. She preferred the smooth B-spline approximation for planning work, presentations, and public communication, as it results in natural surfaces and people are more familiar with continuous regions than with blocks. However, she would use nearest-neighbor interpolation for comparability with results of other software and for communication with other domain experts.

Our strategies to avoid climbing, diving, and leaking artifacts were well received by all experts. Although these artifacts are common in current flood simulation software, the engineers called it dubious to show such results to a customer. One expert initially perceived leaking artifacts as a visualization feature to indicate seeping of water through leaky barriers. The experts agreed that our visualization without leaking artifacts was unambiguous in that regard. Likewise, they appreciated the removal of climbing and diving artifacts, stating that it leads to a more realistic depiction of the water surface. While climbing just looked wrong, one expert said that diving artifacts could also be misinterpreted as a large approaching wave.

All four experts welcomed our visualization of flow directions and high velocities with animated waves and foam. They immediately understood that the effects were deliberately exaggerated and were not part of the flood simulation. They called the use of waves and foam as visual metaphors aesthetic and intuitive and said they were good indicators of the flow strength and direction. All experts were able to quickly identify principal flow directions in an already inundated area from an overview perspective. Additionally, they correctly identified turbulence and a rare case where water against intuition was flowing uphill because of its high velocity. When shown arrow glyphs instead of waves to visualize the velocities, one expert stated they were useful for still images, but she would prefer the waves for animation and videos.

4.9.2 Online Survey

For a quantitative evaluation of our proposed techniques, we conducted an online survey among 96 participants.

Participants We reached out to experts in various fields of work related to flood management with the help of our collaboration partners and provided them with a questionnaire. As one important task of flood management is public communication, we also evaluated our results with members of the general public. Specifically, we asked the users of the commercial flood alert system PegelAlarm [Peg] for participation, assuming that this user group consists of people particularly concerned about floods. For the presentation of the survey results, we grouped the domain experts roughly by profession, for which we asked them to state their profession and areas of responsibility in the survey.
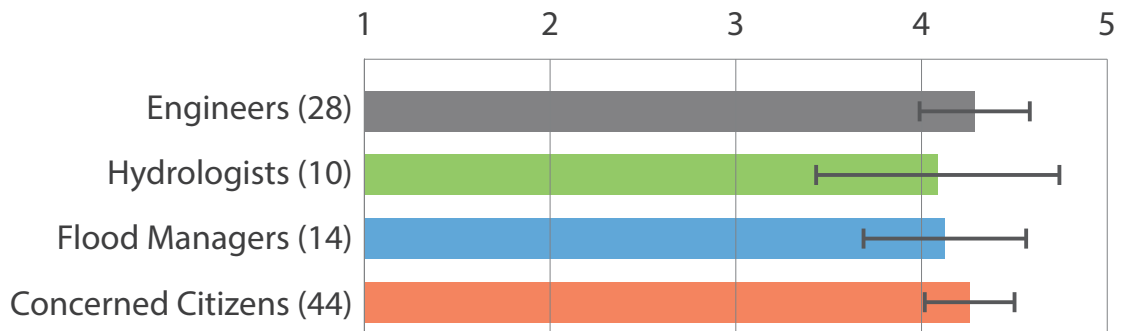
Figure 4.15: Average rating results per group of participants with 95 % confidence intervals.

In summary, the 96 participants included 28 experts working in civil and hydraulic engineering, 10 scientists in the field of hydrology, 14 flood management experts of public authorities, and 44 concerned citizens (see Figure 4.15).

Questionnaire   We asked all participants of our survey to fill out a questionnaire that is still available online [Sura]. A non-interactive version of this questionnaire is provided in Appendix A. As most of the invited participants work in Central Europe, the questionnaire has been distributed in both German and English. It contained eight short videos (37 s to 1 min 55 s) showing visualizations of real-world scenarios within Visdom. For each video, the participants were asked to focus on the flood visualization, which is why almost all user interface elements of the application were hidden. In detail, the videos showed river flooding in rural and urban regions (**Q1**), heavy rains and stormwater runoff (**Q2**), flooding caused by floodwall overtopping (**Q3**), by a floodwall breach (**Q4**), and by dike breaks (**Q5**), the interactive planning of a short-term object protection with sandbags (**Q6**), the interactive planning of long-term protection measures with mobile floodwalls and retention basins (**Q7**), and the impact of a tsunami on a city (**Q8**). After each video, the participants assessed the visualization of the flooding on a scale of 1 (very bad) to 5 (very good). The questionnaire distributed among the domain experts additionally asked to state the relevance of the visualization for their work-related areas of responsibility on a scale of 1 (not relevant) to 5 (very relevant). Below each video, additional comments could be provided.

Results   The detailed results of the survey were provided with the original paper and are available in Appendix B as well as online [Surb]. In Figure 4.15, we give an overview of the groups of participants and their average rating over all questions. On average, our visualizations were well received by all groups and were rated *good*. The favorable ratings of both domain experts and concerned citizens suggest a high suitability of our visualizations for both technical tasks and public communication.

In Figure 4.16, we relate the average rating of each question to the average relevance
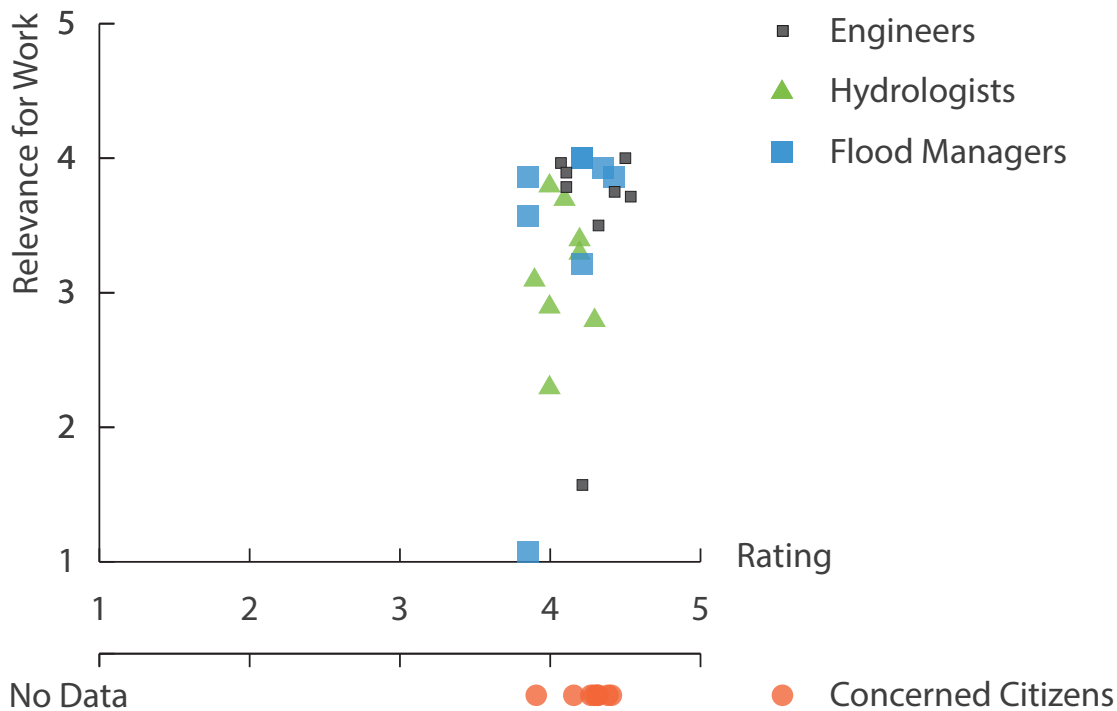
Figure 4.16: Evaluation results per group of participants. Each data point represents the average result of one question within the group.

for the participants' work, separated by group. For the group of concerned citizens, we collected no data on the relevance for their work. Engineers and flood managers assessed the visualizations as both *good* and *relevant* for their work, which we see as a particularly important result of this evaluation, as the experts in these fields are the primarily intended users of our proposed techniques. As most of the scenarios included in the survey deal with flood management tasks, the hydrologists expectedly considered not all of them relevant for their work, but still good. As most of our participants are located in Central Europe, the relevance of one scenario was considered very low by all groups, which is the modeling of a tsunami (**Q8**). Yet, even visualizations not particularly relevant have been rated highly by the participants.

Figure 4.17 shows the average relevance and rating by all experts for each question. This reveals that the visualizations in the context of planning tasks (**Q6** and **Q7**) have been rated best. We attribute this to the interactivity shown in the videos, which demonstrates the benefit of expressive water visualization for interactive tasks.

Besides the quantitative evaluation, the additional comments by some of the participants of the survey provide valuable qualitative statements. In general, the participants received our visualizations very well and praised them as aesthetic, realistic, and descriptive. Eight participants saw a benefit of our visualizations for specific applications, most of all for
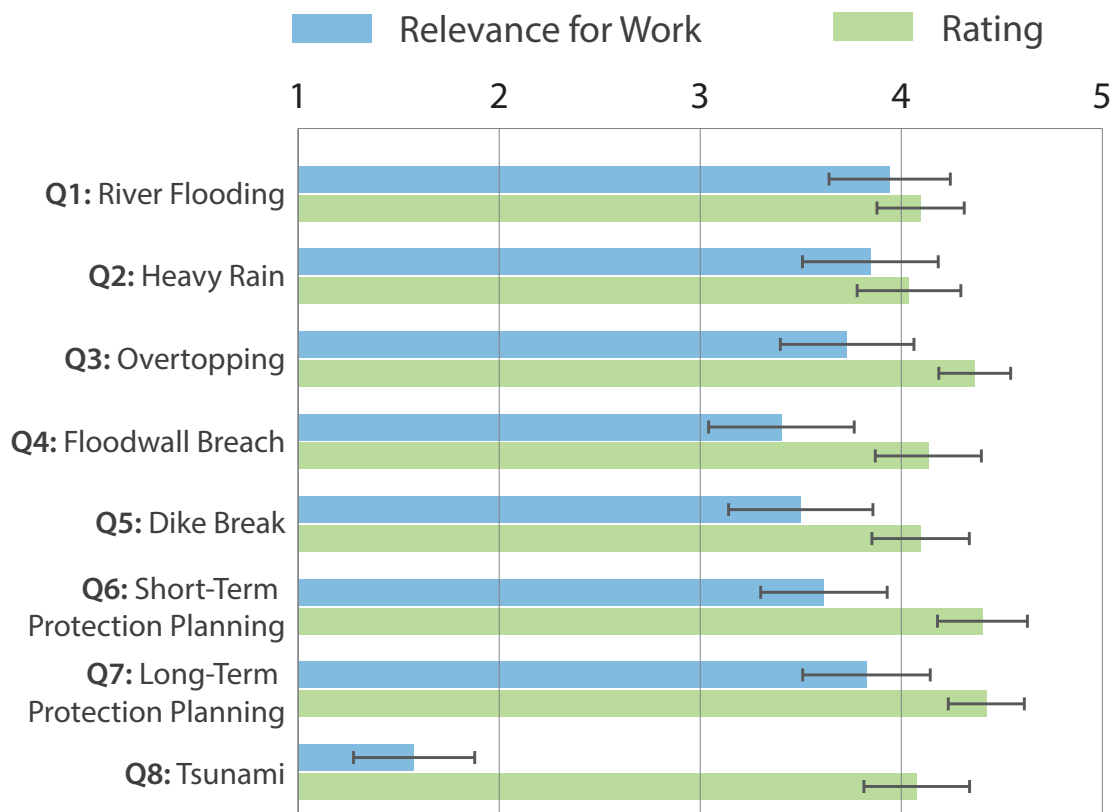
Figure 4.17: Evaluation results per question averaged over the results of all domain experts with 95 % confidence intervals.

public communication, but also for planning and coordination. One participant saw little benefit of our visualization over established 2D methods. 13 participants criticized the chosen color of water and the lack of contrast to the terrain color in the videos. While these colors can be changed freely by the user within our application (for example with transfer-function-based coloring, as demonstrated in Figure 4.18), we consider the study of proper default values an important aspect for future research.

Three participants explicitly stated that the waves and foam helped them identifying flow directions, one participant still could not identify them all the time. One participant made us aware that on mobile devices, the flow directions were not discernible at all, which is an issue we have to address in the future. Six participants praised the combination of the visualization of water with other visualization techniques, in particular the coloring of buildings according to their inundation. Nine participants mentioned combinations with other visualizations that could improve the result, such as driftwood, street names, detailed buildings, and a time line. Our application supports most of these visualizations, which have not been shown in order to draw the participants' attention to the flood visualization for evaluation. Yet, synergies between visualization techniques and
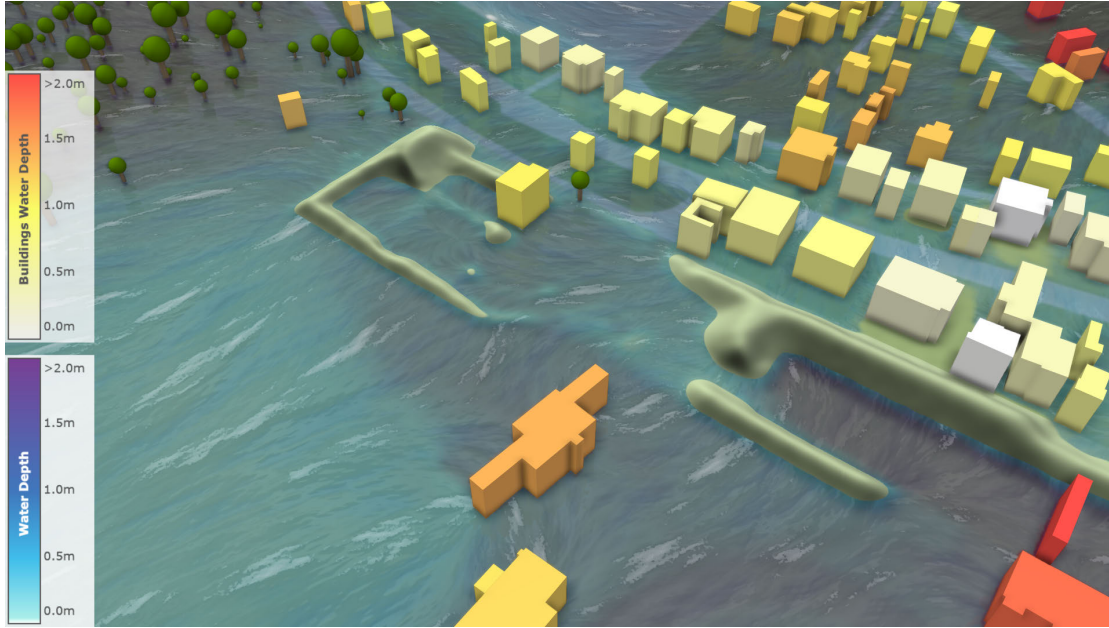
Figure 4.18: Coloring of water and buildings by water depth.

considerations of which ones to always show are aspects that we will further investigate.

4.10 Conclusions and Future Work

In flood management and civil engineering, flood simulation is widely used for the examination of possible floods and their mitigation measures as well as for water-aware urban construction planning. During live sessions with our application partners, situations emerged where the visualizations of such simulation results were misinterpreted due to interpolation artifacts. In this chapter, we present a real-time technique to visualize adaptive water height fields without misleading interpolation artifacts. The adaptive reconstruction scheme treats different levels of the adaptive grid as separate height fields that are combined into a globally C^1-continuous height field. It is efficient enough to be used in interactive applications even for very large data sets. Thus, it fills the gap between fast linear interpolation and slow smooth interpolation. The removal of artifacts counters shortcomings of grid-based interpolation of water height fields by two means. First, by extrapolating the water surface at dry/wet boundaries in a physically plausible way, and, second, by incorporating detailed geometric information in the surface reconstruction that is not available in the regular height field data. The interpolated height fields are rendered efficiently with a continuous level of detail, exploiting hardware tessellation. A visualization of important flow properties by waves and foam is applied to the water surface to support the interpretation of simulation results. The resulting visualizations have been positively evaluated by domain experts and the general public.

Currently, our proposed surface reconstruction is restricted to data structures with at most one level difference between neighboring cells. Future work is necessary to investigate how to remove this restriction. After reconstruction with bicubic interpolation or approximation, height field structures not aligned with the interpolation grid often exhibit staircase artifacts. These artifacts might be reduced by using rotationally invariant reconstruction filters instead of a dimension-wise application of the cubic filter. We therefore consider the combination of fast bicubic interpolation and the computationally more expensive application of thin plate splines a possible direction for future research. The feedback provided by the participants of our survey indicates a varying effectiveness of preset visualization parameters as well as specific combinations of visualization techniques. This issue also requires further research and evaluation.

Conclusions

5.1 Summary

In this thesis, we address the challenges of interactively visualizing large-scale, time-dependent, and uncertain data in the context of flood management. We developed and evaluated task- and user-specific visualization techniques to assist domain experts in their daily work of handling, interpreting, and navigating through these data. Given the vast range of diverse tasks summarized under the term flood management, however, we had to focus on just a selection of tasks that our application partners faced and that we singled out as particularly pressing issues. The interactive analysis and visualization of simulation data in general is still far from being resolved. Still, we have shown that our task-specific solutions support domain experts in their daily work.

The visualization of object-centered vulnerability discussed in Chapter 2 addresses the problem of isolating information about a particular building from an ensemble simulation. The automatic aggregation of heterogeneous data is hidden from the user and relieves her or him of the need to have any knowledge about the data. This allows both the general public and domain experts to investigate flooding results with a specific task in mind. The visualization of vulnerability allows for the straightforward identification of dangerous scenarios and can serve as a starting point for planning protection measures on a larger scale. Additionally, the visualization of flooding impacts on a building can assist in the planning of protection measures on a smaller scale, such as small floodwalls on private property or flood-resistant windows. The proposed visualizations are embedded in the geospatial context to make them as comprehensible as possible. Yet, during the evaluation with domain experts, the indication of uncertainty was still considered too complex for the general public. Nevertheless, the experts agreed that for their own work, our visualizations allow them to work with the time-dependent, uncertain, and high-dimensional simulation data more easily.

The close collaboration with flood managers from Cologne also revealed the difficulties of logistics planning in case of flooding and partially inundated streets, which we addressed in Chapter 3. Here, the underlying time-dependent data are provided by a logistics simulation on street networks and are again aggregated over time to create static flow maps of the entire material transport. The proposed force-driven layout algorithm enables the automatic creation of interactive flow maps in the style of manually drawn historic maps, which offer a simplified, abstract visualization of material flows. Our main contribution to the field of geospatial visualization is the consideration of multiple flows of different materials or more abstract quantities in a single visualization. The logistics experts welcomed this extension, which allows them to identify which construction materials have to be stored in which material depots, for example. We also applied this visualization technique to data from agent-based evacuation simulation, which results in a tidier and more comprehensible visualization than path lines, for example. The application to heavy rain simulation data was not as successful, as the principal flow directions could not be captured properly. An application to uncertain flood simulation data with minimum, average, and maximum water flows as separate components was likewise considered rather experimental than useful in practice.

Obviously, the direct visualization of flood simulation data by means of a closed water surface is very important for flood management, as actions have to be deduced from the expected water extents. Discussions and live sessions with our research and application partners in the past have shown that a natural appearance of the water surface is crucial for their work. This includes planning and validation tasks, but also public communication. First, results with too simple interpolation are perceived as dubious and untrustworthy, and second, simple interpolation can lead to visual artifacts and, subsequently, to wrong interpretations of the visualization. We addressed these issues in Chapter 4 with an emphasis on interactivity, because existing solutions cannot be applied in real time. Our proposed surface reconstruction results in a C^1-continuous surface—even when using adaptive grids—and is tailored to the GPU data model, which enables a highly parallel execution. We applied strategies to reduce visual artifacts stemming from interpolation that we have identified in previous work. Finally, we proposed the shading of the water surface with animated waves and foam as visual metaphors to visualize flow properties. The results of the direct visualization of flood simulation data have been highly praised by our application partners as well as by a large group of domain experts from various fields.

5.2   Retrospect

The visual quality and overall usefulness of the visualization techniques presented in this thesis have been highly rated by our application partners and other domain experts. The aimed at app-like concept of task-specific solutions in a modular system as opposed to a comprehensive all-at-once visualization has proven to be an expedient strategy. Since the beginning of this thesis in late 2014, we have incrementally added novel visualization techniques to the existing decision support system Visdom to address pressing issues that

Figure 5.1: Application of visualization techniques in different contexts. (a) Adaptive cutaways applied to buildings to avoid occlusion of a floodwall selected by the user. (b) Terrain cutaway used to excavate sewer pipes for the inspection of sewer overflows.

our partners faced in their daily work. With the rapid advances in flood modeling and simulation hardware, both the requirements for their work and the developed techniques have evolved over time. The choice of a modular design—even though born out of necessity as a holistic solution seemed utopian—allowed us to adapt and repurpose developed techniques as needed. New use cases for the visualization techniques have emerged that we did not anticipate at the time of publication.

For example, in Chapter 2, we discussed the use of adaptive cutaways for buildings to avoid occlusion of the building of interest selected by the user, while still providing enough context for orientation in the scene. This technique can be generalized to interactively hide occluders of arbitrary important objects, structures, or positions, like the floodwall selected for inspection by the user in Figure 5.1a. Similarly, the terrain cutaways introduced in Chapter 2 have been used to virtually excavate buildings to reveal their cellar walls for inspection. With the development of a sewer network simulation in recent years, the inspection of simulation results defined on the underground sewer network has become an important task. We adopted terrain cutaways for this new use case and allow the user to interactively sketch an area of interest on the terrain, after which the underground geometry is revealed. This is shown in Figure 5.1b, where a sewer overflow indicated with a red circle is inspected.

The flow maps introduced in Chapter 3 for material flow have also proven useful for various use cases. In particular, they can be used as a more abstract visualization of rivers and streams as compared to actual water surfaces, which is especially helpful when investigating river networks from an overview perspective. Individual streams can be represented by individual flow arrows with colors indicating, for example, a unique ID or water discharges, as shown in Figure 5.2a. Here, the displayed arrows allow the user to instantly discern flow directions of the streams. For the simulation of river flooding on entire river networks, it is necessary that all individual streams are correctly inserted and linked in the network structure. To verify these external river data, we use a data validation application in which flow maps visualize the hierarchical relationship of the streams, as shown in Figure 5.2b. Here, the actual river geometry drawn in blue is overlaid with the river network topology, where green flow map arrows pointing upstream towards the river head indicate valid river segments.

Recalling the aim of this thesis formulated in Section 1.3, we ventured out to tame the nearly unmanageably large uncertain and time-dependent simulation data in an interactive system, while also isolating and communicating the important information and its uncertainty in a comprehensible and aesthetic way. We have achieved most, but not all of this ambitious goal. Considering the runtime performance, we have developed algorithms and data structures facilitating parallelization on the GPU that make it possible to handle the massive simulation and geospatial data at interactive framerates. Furthermore, in all relevant visualization techniques, the time dependence of the simulation data could be eliminated by aggregation in order to obtain still images and reduce the complexity of the resulting visualizations, for example with flow maps of the entire logistics operation.
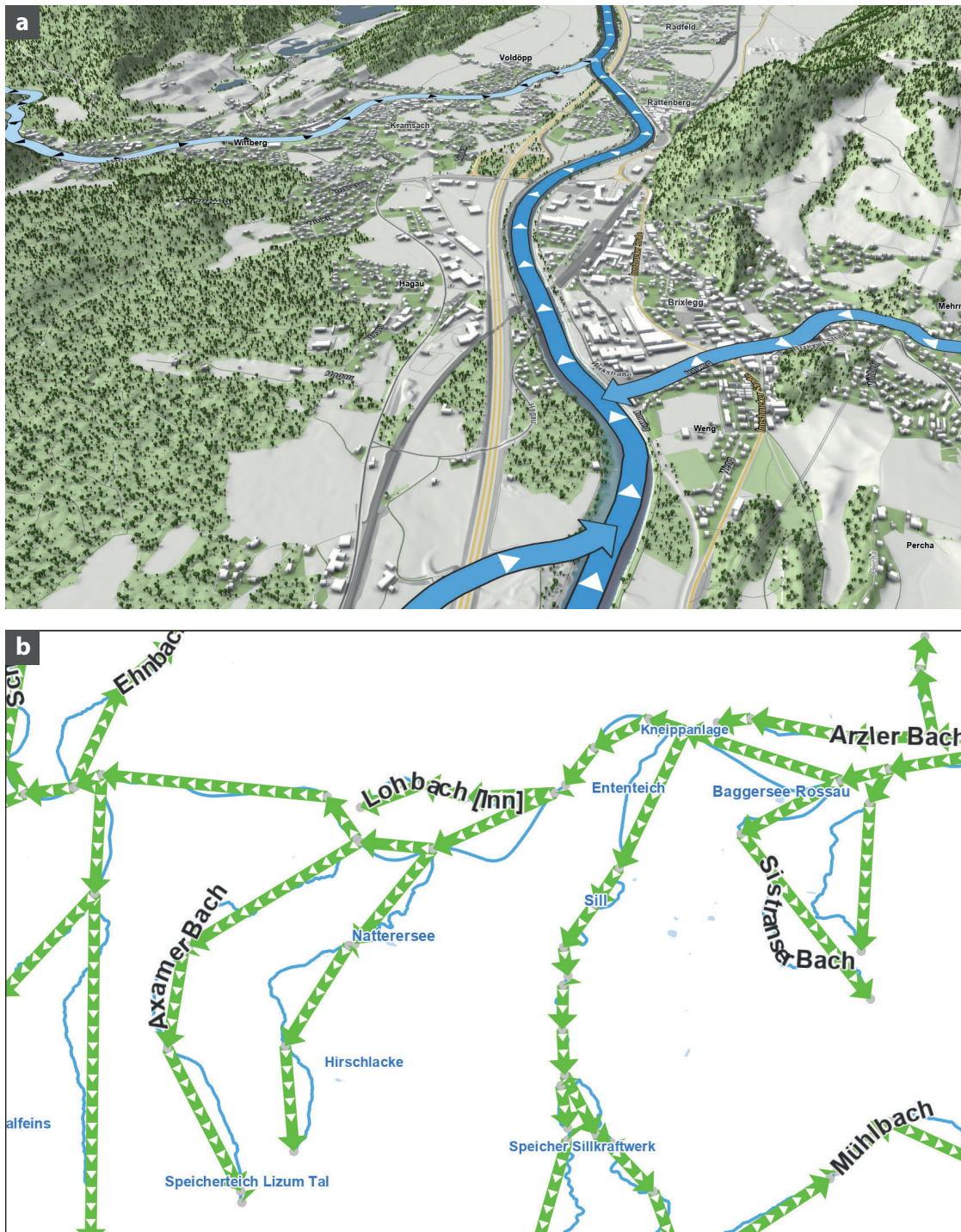
Figure 5.2: Application of flow maps in different contexts. (a) Visualization of rivers and streams for the identification of flow directions. (b) Visualization of the topology (green) of a river network (blue) for the validation of input data.

The treatment of uncertainty, however, did not work out as planned. While our application partners welcomed the visualization of uncertain vulnerabilities and flooding impacts discussed in Chapter 2, they suspected it would be too complex to be used for public communication. The indication of uncertainty in flow maps was met with even more skepticism, meaning the domain experts did not even see a benefit of this visualization for their own work. The visualization of uncertainty remains immensely difficult, especially when trying to integrate this rather abstract information into the geospatial context.

However, the biggest challenge we faced during the course of this thesis was the development of a proper and efficient direct visualization of water surfaces and its subsequent publication as scientific work. Starting with the addition of simple surface reflections as an initial assessment of the author for the PhD position in 2014 and the presentation of our results summarized in Chapter 4 at a scientific conference in 2019, the visualization of water surfaces has been the encompassing and most pervasive topic of this thesis. The reasons for this are manifold. First of all, water visualization is hard from a technical point of view. The appearance of a water surface in real life is a combination of countless complex interactions of light with water and other molecules at and below the surface, which is hard to model in a renderer than is not physically-based, and of course also computationally very expensive. This goes hand in hand with the fact that water is ubiquitous and every sighted person has a distinct idea of how water looks and behaves. Even slight imperfections in the water appearance, such as the visual artifacts we addressed in Section 4.5, are very noticeable, which makes the development of a fast visualization relying on a lot of approximations a very unforgiving task. While the work on this topic started at a time when only data defined on regular grids were used, the welcomed advances in graphics hardware made it possible to operate on much larger simulation domains. Thus, the need to handle adaptive grids emerged, which complicated the task of creating water surface geometry additionally to the existing challenge of its visualization. After at least seven complete iterations of the rendering system, including several novel approaches to manual height field tessellation, mesh displacement, and real-time ray tracing, we arrived at the state presented in Chapter 4, which still has its limitations. From a scientific point of view, publishing our results in a renowned journal has been just as much of a challenge. There exists a common understanding that water rendering in real time is "solved", because it appears like this in the many astonishingly realistic and aesthetic video games in recent years. In reality, water bodies in video games are carefully crafted by artists, which are not bound by physical plausibility or dynamic data, which removes almost all of the challenges we faced. While mimicking traditional, hand-crafted art with algorithms (such as the hand-drawn flow map in Section 3.2) is an established procedure in the field of visualization, mimicking digital art seems to be a rather unconventional idea that requires substantial justification in a scientific context.

5.3   Outlook

There is still a lot of work to be done. In the context of the individual task-specific visualizations, we have singled out open questions and starting points for possible future

work in Section 2.11, Section 3.8, and Section 4.10. More fundamentally, we see the comprehensive visualization of uncertainty in the geospatial domain as a major issue in the future, as uncertain data will become more prevalent as ensemble simulations become more feasible. In particular, the communication of uncertain predictions to non-experts is a large challenge, as simplified visualizations, such as only communicating the worst-case outcome, also have their downsides.

A second issue we see, is the unknown scalability of our proposed techniques regarding simulation data size. While we target requirements and a workflow that are common nowadays, it is hard to predict how the field of interactive decision support in flood management will evolve. Even for large simulation domains, it is currently possible to simulate, process, and visualize flooding scenarios interactively on a high-end computer. Using cloud-computing for massive simulation ensembles, for example, would significantly increase the simulation data and possibly require new strategies for visualization, for example distributed or out-of-core rendering, which would introduce various new challenges for interactivity.

It is also an open question how well the system we describe would scale in a multi-user setting. The main application of Visdom is an expert system designed for engineers and flood managers. However, in Chapter 2, we already proposed user-specific apps for different user groups, including the general public, which would ideally be made available online as an alternative to static flood risk maps. This would allow home owners and other concerned citizens to use a simplified decision support system online, inspect flood risks, test local protection measures for their own homes, and then take actions. To remain interactive in such a setting, the system would likely have to be able to cache and reuse simulation, aggregation, and visualization results among different users, which would require an intelligent mechanism for incremental simulation and visualization.

We have the best prospects that these open problems will be tackled and overcome in the future. As already mentioned in Section 1.1.2, 3D visualization is becoming an established tool in flood management and the number of interactive decision support systems is growing. The acceptance of this new technology by flood management experts will increase over time, which will not only lead to more practical applications for these systems, but will also highlight limitations of current systems and approaches, which will then amplify the research for better solutions.

Bibliography

[3Di]   3Di Water Management. `http://3diwatermanagement.com/` (last visited on January, 18$^{\text{th}}$ 2020).

[AA11]   Natalia Andrienko and Gennady Andrienko. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):205–219, 2011.

[AAB⁺13]   Gennady Andrienko, Natalia Andrienko, Peter Bak, Daniel Keim, and Stefan Wrobel. *Visual Analytics of Movement*. Springer Science & Business Media, 2013.

[AABW12]   Gennady Andrienko, Natalia Andrienko, Michael Burch, and Daniel Weiskopf. Visual analytics methodology for eye movement studies. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2889–2898, 2012.

[AB05]   Emmanuel Audusse and Marie-Odile Bristeau. A well-balanced positivity preserving "second-order" scheme for shallow water flows on unstructured meshes. *Journal of Computational Physics*, 206(1):311–333, 2005.

[ARMN16]   Sam Amirebrahimi, Abbas Rajabifard, Priyan Mendis, and Tuan Ngo. A framework for a microscale flood damage assessment and visualization for a building using BIM–GIS integration. *International Journal of Digital Earth*, 9(4):363–386, 2016.

[Bel04]   Vladimir Belyaev. Real-time rendering of shallow water. In *GraphiCon Proceedings*, pages 1–6. GraphiCon Scientific Society, 2004.

[BF08]   Michael Burns and Adam Finkelstein. Adaptive cutaways for comprehensible rendering of polygonal scenes. *ACM Transactions on Graphics*, 27(5):154:1–7, 2008.

[BHMF08]   Johanna Beyer, Markus Hadwiger, Torsten Möller, and Laura Fritz. Smooth mixed-resolution GPU volume rendering. In *Proceedings of the Fifth Eurographics / IEEE VGTC Conference on Point-Based Graphics*, pages 163–170. Eurographics Association, 2008.

[BHN⁺19]   Andreas Buttinger-Kreuzhuber, Zsolt Horváth, Sebastian Noelle, Günter Blöschl, and Jürgen Waser. A fast second-order shallow water scheme on two-dimensional structured grids over abrupt topography. *Advances in Water Resources*, 127:89–108, 2019.

[BHP⁺17]   Günter Blöschl, Julia Hall, Juraj Parajka, Rui A. P. Perdigão, Bruno Merz, Berit Arheimer, Giuseppe T. Aronica, Ardian Bilibashi, Ognjen Bonacci, Marco Borga, Ivan Čanjevac, Attilio Castellarin, Giovanni B. Chirico, Pierluigi Claps, Károly Fiala, Natalia Frolova, Liudmyla Gorbachova, Ali Gül, Jamie Hannaford, Shaun Harrigan, Maria Kireeva, Andrea Kiss, Thomas R. Kjeldsen, Silvia Kohnová, Jarkko J. Koskela, Ondrej Ledvinka, Neil Macdonald, Maria Mavrova-Guirguinova, Luis Mediero, Ralf Merz, Peter Molnar, Alberto Montanari, Conor Murphy, Marzena Osuch, Valeryia Ovcharuk, Ivan Radevski, Magdalena Rogger, José L. Salinas, Eric Sauquet, Mojca Šraj, Jan Szolgay, Alberto Viglione, Elena Volpi, Donna Wilson, Klodian Zaimi, and Nenad Živković. Changing climate shifts timing of European floods. *Science*, 357(6351):588–590, 2017.

[BLE⁺14]   Marina Brovka, José Iván López, José María Escobar, José Manuel Cascón, and Rafael Montenegro. Construction of polynomial spline spaces over quadtree and octree T-meshes. *Procedia Engineering*, 82:21–33, 2014.

[BMA10]   Alex Beutel, Thomas Mølhave, and Pankaj K. Agarwal. Natural neighbor interpolation based grid DEM construction using a GPU. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 172–181. ACM, 2010.

[BMPM12]   Maryam Booshehrian, Torsten Möller, Randall M. Peterman, and Tamara Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. *Computer Graphics Forum*, 31(3pt3):1235–1244, 2012.

[Bob08]   Tom Bobach. *Natural Neighbor Interpolation - Critical Assessment and New Contributions*. PhD thesis, TU Kaiserslautern, 2008.

[BOL12]   Ken Brodlie, Rodolfo Allendes Osorio, and Adriano Lopes. A review of uncertainty in data visualization. In *Expanding the Frontiers of Visual Analytics and Visualization*, pages 81–109. Springer, 2012.

[Bon11]   Xavier Bonaventura. Terrain and ocean rendering with hardware tessellation. In Wolfgang Engel, editor, *GPU Pro 2*, pages 3–14. A K Peters, 2011.

[Boo]   The Boost Graph Library. `http://www.boost.org/doc/libs/release/libs/graph/` (last visited on January, 18th 2020).

[BOR14]   Richard K. Beatson, Wen Eng Ong, and Igor Rychkov. Faster fast evaluation of thin plate splines in two dimensions. *Journal of Computational and Applied Mathematics*, 261:201–212, 2014.

[Bow13]   Huw Bowles. Oceans on a shoestring: Shape representation, meshing and shading. Advances in Real-Time Rendering in Games II course, ACM SIGGRAPH, 2013.

[BRK08]   Günter Blöschl, Christian Reszler, and Jürgen Komma. A spatially distributed flash flood forecasting model. *Environmental Modelling & Software*, 23:464–478, 2008.

[BST15]   Dmitry Borovikov, Igor V. Sokolov, and Gábor Tóth. An efficient second-order accurate and continuous interpolation for block-adaptive grids. *Journal of Computational Physics*, 297:599–610, 2015.

[BSV11]   Kevin Buchin, Bettina Speckmann, and Kevin Verbeek. Flow map layout via spiral trees. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2536–2544, 2011.

[BSV15]   Kevin Buchin, Bettina Speckmann, and Kevin Verbeek. Angle-restricted Steiner arborescences for flow map layout. *Algorithmica*, 72(2):656–685, 2015.

[BSW10]   Florian Bagar, Daniel Scherzer, and Michael Wimmer. A layered particle-based fluid model for real-time rendering of water. *Computer Graphics Forum*, 29(4):1383–1389, 2010.

[CBK+19]   Daniel Cornel, Andreas Buttinger-Kreuzhuber, Artem Konev, Zsolt Horváth, Michael Wimmer, Raimund Heidrich, and Jürgen Waser. Interactive visualization of flood and heavy rain simulations. *Computer Graphics Forum*, 38(3):25–39, 2019.

[CG14]   Michael Correll and Michael Gleicher. Error bars considered harmful: Exploring alternate encodings for mean and error. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2142–2151, 2014.

[Che13]   Tangpei Cheng. Accelerating universal kriging interpolation algorithm using CUDA-enabled GPU. *Computers & Geosciences*, 54:178–183, 2013.

[CKS+15]   Daniel Cornel, Artem Konev, Bernhard Sadransky, Zsolt Horváth, Eduard Gröller, and Jürgen Waser. Visualization of object-centered vulnerability to possible flood hazards. *Computer Graphics Forum*, 34(3):331–340, 2015.

[CKS+16]   Daniel Cornel, Artem Konev, Bernhard Sadransky, Zsolt Horváth, Andrea Brambilla, Ivan Viola, and Jürgen Waser. Composite flow maps. *Computer Graphics Forum*, 35(3):461–470, 2016.

[CM10]   Nuttapong Chentanez and Matthias Müller. Real-time simulation of large
            bodies of water with small scale details. In *Proceedings of the ACM SIG-
            GRAPH / Eurographics Symposium on Computer Animation*, pages 197–206.
            Eurographics Association, 2010.

[Col07]   Mat Collins. Ensembles and probabilities: a new era in the prediction of
            climate change. *Philosophical Transactions of the Royal Society A: Mathe-
            matical, Physical and Engineering Sciences*, 365(1857):1957–1970, 2007.

[Com]   Composite Flow Maps Video. `http://visdom.at/media/videos/`
            `mp4/composite_flow_maps.mp4` (last visited on January, 18th 2020).

[COT+11]   Luciano da Fontoura Costa, Osvaldo N. Oliveira Jr., Gonzalo Travieso, Fran-
            cisco Aparecido Rodrigues, Paulino Ribeiro Villas Boas, Lucas Antiqueira,
            Matheus Palhares Viana, and Luis Enrique Correa da Rocha. Analyzing
            and modeling real-world phenomena with complex networks: a survey of
            applications. *Advances in Physics*, 60(3):329–412, 2011.

[CP09]   Hannah Louise Cloke and Florian Pappenberger. Ensemble flood forecasting:
            A review. *Journal of Hydrology*, 375(3-4):613–626, 2009.

[Csé19]   Balázs Csébfalvi. Beyond trilinear interpolation: Higher quality for free.
            ACM Transactions on Graphics, 38(4):56:1–8, 2019.

[CZQ+08]   Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li.
            Geometry-based edge clustering for graph visualization. *IEEE Transactions
            on Visualization and Computer Graphics*, 14(6):1277–1284, 2008.

[DB12]   Jonathan Dupuy and Eric Bruneton. Real-time animation and rendering of
            ocean whitecaps. In *SIGGRAPH Asia Technical Briefs*, pages 15:1–3. ACM,
            2012.

[DCL+08]   Jiansong Deng, Falai Chen, Xin Li, Changqi Hu, Weihua Tong, Zhouwang
            Yang, and Yuyu Feng. Polynomial splines over hierarchical T-meshes. *Graph-
            ical models*, 70(4):76–86, 2008.

[D'E07]   Claudia D'Eliso. *Breaching of sea dikes initiated by wave overtopping. A
            tiered and modular modelling approach.* PhD thesis, Faculty of Architecture,
            Civil Engineering and Environmental Sciences, Institute of Technology, Uni-
            versity of Braunschweig and Faculty of Engineering, University of Florence,
            2007.

[DMW+01]   Aiguo Dai, Gerald A. Meehl, Warren M. Washington, Tom M. L. Wigley,
            and Julie M. Arblaster. Ensemble simulation of twenty-first century cli-
            mate changes: Business-as-usual versus CO_2 stabilization. *Bulletin of the
            American Meteorological Society*, 82(11):2377–2388, 2001.

[DNCP10]   David Demeritt, Sébastien Nobert, Hannah Cloke, and Florian Pappenberger. Challenges in communicating and using ensembles in operational flood forecasting. *Meteorological applications*, 17(2):209–222, 2010.

[DSD14]   Alberto Debiasi, Bruno Simões, and Raffaele De Amicis. Supervised force directed algorithm for the generation of flow maps. In *Proceedings of the 22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 193–202. Union Agency, 2014.

[DV10]   Urška Demšar and Kirsi Virrantaus. Space–time density of trajectories: exploring spatio-temporal patterns in movement data. *International Journal of Geographical Information Science*, 24(10):1527–1542, 2010.

[EHP+11]   Ozan Ersoy, Christophe Hurter, Fernando V. Paulovich, Gabriel Cantareiro, and Alexandru Telea. Skeleton-based edge bundling for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2364–2373, 2011.

[Eur07]   The European Parliament and the Council of the European Union. Directive 2007/60/EC of the European Parliament and of the Council of 23 October 2007 on the assessment and management of flood risks (Text with EEA relevance). Official Journal of the European Union, L 288, 2007. `https://eur-lex.europa.eu/eli/dir/2007/60/oj` (last visited on January, 18th 2020).

[FBHD17]   Franz G. Fuchs, Oliver J. D. Barrowclough, Jon M. Hjelmervik, and Heidi E. I. Dahl. Direct interactive visualization of locally refined spline volumes for scalar and vector fields. *arXiv e-prints*, abs/1707.01170:1–11, 2017.

[FBS+12]   Christopher B. Field, Vicente Barros, Thomas F. Stocker, Qin Dahe, David Jon Dokken, Kristie L. Ebi, Michael D. Mastrandrea, Katharine J. Mach, Gian-Kasper Plattner, Simon K. Allen, Melinda Tignor, and Pauline M. Midgley, editors. *Managing the Risks of Extreme Events and Disasters to Advance Climate Change Adaptation. A Special Report of Working Groups I and II of the Intergovernmental Panel on Climate Change.* Cambridge University Press, 2012. `https://www.ipcc.ch/site/assets/uploads/2018/03/SREX_Full_Report-1.pdf` (last visited on January, 18th 2020).

[FFGS14]   Jan-Thomas Fischer, Reinhard Fromm, Peter Gauer, and Betty Sovilla. Evaluation of probabilistic snow avalanche simulation ensembles with Doppler radar observations. *Cold Regions Science and Technology*, 97:151–158, 2014.

[FGM+11]   Mark A. Finney, Isaac C. Grenfell, Charles W. McHugh, Robert C. Seli, Diane Trethewey, Richard D. Stratton, and Stuart Brittain. A method for ensemble wildland fire simulation. *Environmental Modeling & Assessment*, 16(2):153–167, 2011.

[Fin04]   Mark Finch. Effective water simulation from physical models. In Randima Fernando, editor, *GPU Gems*, pages 5–29. Addison-Wesley, 2004.

[Flo]   FloodViz – Visual Analytics for Assessment and Interpretation of Simulated River Flooding. `https://www.gri.msstate.edu/research/floodviz/` (last visited on January, 18th 2020).

[Fra82]   Richard Franke. Smooth interpolation of scattered data by local thin plate splines. *Computers & Mathematics with Applications*, 8(4):273–281, 1982.

[GH12]   Carlos Gonzalez-Ochoa and Doug Holder. Water technology of Uncharted. Game Developers Conference, 2012.

[GJAG14]   Eduardo Gutiérrez de Ravé, Francisco J. Jiménez-Hornero, Ana B. Ariza-Villaverde, and Juán M. Gómez-López. Using general-purpose computing on graphics processing units (GPGPU) to accelerate the ordinary kriging algorithm. *Computers & Geosciences*, 64:1–6, 2014.

[Gon16]   Carlos Gonzalez-Ochoa. Rendering rapids in Uncharted 4. Advances in Real-Time Rendering in Games I course, ACM SIGGRAPH, 2016.

[Goo00]   Pierre Goovaerts. Geostatistical approaches for incorporating elevation into the spatial interpolation of rainfall. *Journal of Hydrology*, 228(1):113–129, 2000.

[GR05]   Tilmann Gneiting and Adrian E. Raftery. Weather forecasting with ensemble methods. *Science*, 310(5746):248–249, 2005.

[Gri11]   Bronwen Grimes. Making and using non-standard textures: Manipulating UVs through color data in Portal 2. Game Developers Conference, 2011.

[GSH+15]   Sebastian Grottel, Joachim Staib, Torsten Heyer, Benjamin Vetter, and Stefan Gumhold. Real-time visualization of urban flood simulation data for non-professionals. In *Workshop on Visualisation in Environmental Sciences (EnvirVis)*, pages 37–41. Eurographics Association, 2015.

[Guo09]   Diansheng Guo. Flow mapping and multivariate visualization of large spatial interaction data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1041–1048, 2009.

[GWL07]   Kees De Gooijer, Ludolph R. Wentholt, and Erik Jan Langkamp. FLIWAS: Flood information and warning system. In *Proceedings of the 5th Annual Mekong Flood Forum*, pages 163–170. Mekong River Commission, 2007.

[Har96]   Robert L. Harris. *Information graphics: A comprehensive illustrated reference.* Oxford University Press, 1996.

[HBA$^+$99] Mike Hulme, Elaine M. Barrow, Nigel W. Arnell, Paula A. Harrison, Timothy C. Johns, and Thomas E. Downing. Relative impacts of human-induced climate change and natural climate variability. *Nature*, 397(6721):688–691, 1999.

[HBK$^+$20] Zsolt Horváth, Andreas Buttinger-Kreuzhuber, Artem Konev, Daniel Cornel, Jürgen Komma, Günter Blöschl, Sebastian Noelle, and Jürgen Waser. Comparison of fast shallow-water schemes on real-world floods. *Journal of Hydraulic Engineering*, 146(1):05019005:1–16, 2020.

[HET12] Christophe Hurter, Ozan Ersoy, and Alexandru Telea. Graph bundling by kernel density estimation. *Computer Graphics Forum*, 31(3pt1):865–874, 2012.

[HET13] Christophe Hurter, Ozan Ersoy, and Alexandru Telea. Smooth bundling of large streaming and sequence graphs. In *6th IEEE Pacific Visualization Symposium (PacificVis)*, pages 41–48. IEEE, 2013.

[HHL18] Paul Haynes, Sigrid Hehl-Lange, and Eckart Lange. Mobile augmented reality for flood visualisation. *Environmental Modelling & Software*, 109:380–389, 2018.

[HLSH13] Jingming Hou, Qiuhua Liang, Franz Simons, and Reinhard Hinkelmann. A 2D well-balanced shallow flow model for unstructured grids with novel slope source term treatment. *Advances in Water Resources*, 52:107–131, 2013.

[Hol06] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[Hor15] Christopher J. Horvath. Empirical directional wave spectra for computer graphics. In *Proceedings of the Symposium on Digital Production*, pages 29–39. ACM, 2015.

[HPW$^+$16] Zsolt Horváth, Rui A. P. Perdigão, Jürgen Waser, Daniel Cornel, Artem Konev, and Günter Blöschl. Kepler shuffle for real-world flood simulations on GPUs. *The International Journal of High Performance Computing Applications*, 30(4):379–395, 2016.

[HSC$^+$05] Justin Hensley, Thorsten Scheuermann, Greg Coombe, Montek Singh, and Anselmo Lastra. Fast summed-area table generation and its applications. *Computer Graphics Forum*, 24(3):547–555, 2005.

[HSH10] Liang Hu, Pedro V. Sander, and Hugues Hoppe. Parallel view-dependent level-of-detail control. *IEEE Transactions on Visualization and Computer Graphics*, 16(5):718–728, 2010.

[Hut95]   Michael F. Hutchinson. Interpolating mean rainfall using thin plate smoothing splines. *International Journal of Geographical Information Systems*, 9(4):385–403, 1995.

[HVW09]   Danny Holten and Jarke J. Van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009.

[HWP+15]   Zsolt Horváth, Jürgen Waser, Rui A. P. Perdigão, Artem Konev, and Günter Blöschl. A two-dimensional numerical scheme of dry/wet fronts for the Saint-Venant system of shallow water equations. *International Journal for Numerical Methods in Fluids*, 77(3):159–182, 2015.

[JRS12]   Bruno Jobard, Nicolas Ray, and Dmitry Sokolov. Visualizing 2D flows with animated arrow plots. *arXiv e-prints*, abs/1205.5204:1–12, 2012.

[JW15]   Stefan Jeschke and Chris Wojtan. Water wave animation via wavefront parameter interpolation. *ACM Transactions on Graphics*, 34(3):27:1–14, 2015.

[JW17]   Stefan Jeschke and Chris Wojtan. Water wave packets. *ACM Transactions on Graphics*, 36(4):103:1–12, 2017.

[KC14]   Marta Kryven and William Cowan. What does water look like? In *Proceedings of the Workshop on Computational Aesthetics*, pages 53–56. ACM, 2014.

[KdH13]   Christian Kehl and Gerwin de Haan. Interactive simulation and visualisation of realistic flooding scenarios. In Sisi Zlatanova, Rob Peters, Arta Dilo, and Hans Scholten, editors, *Intelligent Systems for Crisis Management: Geoinformation for Disaster Management (Gi4DM)*, pages 79–93. Springer, 2013.

[KH02]   Ralf Kähler and Hans-Christian Hege. Texture-based volume rendering of adaptive mesh refinement data. *The Visual Computer*, 18(8):481–492, 2002.

[Kid03]   David B. Kidner. Higher-order interpolation of regular grid digital elevation models. *International Journal of Remote Sensing*, 24(14):2981–2987, 2003.

[KJCH15]   HyeongYeop Kang, Hanyoung Jang, Chang-Sik Cho, and JungHyun Han. Multi-resolution terrain rendering with GPU tessellation. *The Visual Computer*, 31(4):455–469, 2015.

[KKZ+00]   Tiruvalam N. Krishnamurti, Chandra M. Kishtawal, Zhan Zhang, Timothy LaRow, David Bachiochi, Eric Williford, Sulochana Gadgil, and Sajani Surendran. Multimodel ensemble forecasts for weather and seasonal climate. *Journal of Climate*, 13(23):4196–4216, 2000.

[KLCK17]   Jong Hyun Kim, Jung Lee, Sungdeok Cha, and Chang Hun Kim. Efficient representation of detailed foam waves by incorporating projective space. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2056–2068, 2017.

[Köl]   Stadtentwässerungsbetriebe Köln. `http://www.steb-koeln.de/` (last visited on January, 18th 2020).

[KP07]   Alexander Kurganov and Guergana Petrova. A second-order well-balanced positivity preserving central-upwind scheme for the Saint-Venant system. *Communications in Mathematical Sciences*, 5(1):133–160, 2007.

[KRBH07]   Jürgen Komma, Christian Reszler, Günter Blöschl, and Thomas Haiden. Ensemble prediction of floods – catchment non-linearity and forecast probabilities. *Natural Hazards and Earth System Sciences*, 7:431–444, 2007.

[KT09]   Byungmoon Kim and Panagiotis Tsiotras. Image segmentation on cell-center sampled quadtree and octree grids. *SPIE Wavelet Applications in Industrial Processing VI*, 7248:L:1–9, 2009.

[KZX⁺14]   Sungahn Ko, Jieqiong Zhao, Jing Xia, Shehzad Afzal, Xiaoyu Wang, Greg Abram, Niklas Elmqvist, Len Kne, David Van Riper, Kelly P. Gaither, Shaun Kennedy, William J. Tolone, William Ribarsky, and David S. Ebert. VASA: Interactive computational steering of large asynchronous simulation pipelines for societal infrastructure. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1853–1862, 2014.

[LBA10]   Antoine Lambert, Romain Bourqui, and David Auber. Winding roads: Routing edges into bundles. *Computer Graphics Forum*, 29(3):853–862, 2010.

[LBHS14]   Johannes G. Leskens, Marcela Brugnach, Arjen Y. Hoekstra, and Wytze Schuurmans. Why are decisions in flood disaster management so poorly supported by information from flood models? *Environmental Modelling & Software*, 53:53–61, 2014.

[LCKD16]   Xin Li, FaLai Chen, HongMei Kang, and JianSong Deng. A survey on the local refinable splines. *Science China Mathematics*, 59(4):617–644, 2016.

[LD10]   Haik Lorenz and Jürgen Döllner. 3D feature surface properties and their application in geovisualization. *Computers, Environment and Urban Systems*, 34(6):476–483, 2010.

[Leh18]   Barry Lehrman. Visualizing water infrastructure with Sankey maps: a case study of mapping the Los Angeles Aqueduct, California. *Journal of Maps*, 14(1):52–64, 2018.

[LH04]   Frank Losasso and Hugues Hoppe. Geometry clipmaps: Terrain rendering using nested regular grids. *ACM Transactions on Graphics*, 23(3):769–776, 2004.

[LHV13]   Endre M. Lidal, Helwig Hauser, and Ivan Viola. Design principles for cutaway visualization of geological models. In *Proceedings of the 28th Spring Conference on Computer Graphics*, pages 47–54. ACM, 2013.

[Lia11]   Qiuhua Liang. A structured but non-uniform Cartesian grid-based model for the shallow water equations. *International Journal for Numerical Methods in Fluids*, 66(5):537–554, 2011.

[LJL13]   Hyunjin Lee, Yuna Jeong, and Sungkil Lee. Recursive tessellation. In *ACM SIGGRAPH Asia Posters*, page 16:1. ACM, 2013.

[LKT+17]   Johannes G. Leskens, Christian Kehl, Tim Tutenel, Timothy Kol, Gerwin de Haan, Guus Stelling, and Elmar Eisemann. An interactive simulation and visualization tool for flood analysis usable for practitioners. *Mitigation and Adaptation Strategies for Global Change*, 22(2):307–324, 2017.

[LMMG15]   Asier Lacasta, Mario Morales-Hernández, Javier Murillo, and Pilar García-Navarro. GPU implementation of the 2D shallow water equations for the simulation of rainfall/runoff events. *Environmental Earth Sciences*, 74(11):7295–7305, 2015.

[LNL09]   Jianxin Luo, Guiqiang Ni, and Kun Luo. Projected displaced-mesh: A fast terrain visualization algorithm. In *International Conference on Image Analysis and Signal Processing*, pages 226–232. IEEE, 2009.

[LSGE08]   Yotam Livny, Neta Sokolovsky, Tal Grinshpoun, and Jihad El-Sana. A GPU persistent grid mapping for terrain rendering. *The Visual Computer*, 24(2):139–153, 2008.

[Lum07]   Darren Lumbroso. Review report of operational flood management methods and models. Technical Report T17–07–01, FLOODSite Consortium, 2007. `http://www.floodsite.net/html/partner_area/project_docs/Task17_report_M17_1review_v1_1.pdf` (last visited on January, 18th 2020).

[Lün]   Stadtbetrieb Abwasserbeseitigung Lünen. `https://abwasser-luenen.de/starkregen-2/` (last visited on January, 18th 2020).

[MB96]   Nelson Max and Barry Becker. Flow visualization using moving textures. In *Proceedings of the ICASE/LaRC Symposium on Visualizing Time-Varying Data*, NASA Conference Publication 3321, pages 77–87. NASA, 1996.

128

[MD06]   Stefan Maass and Jürgen Döllner. Dynamic annotation of interactive environments using object-integrated billboards. In *Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 327–334. Union Agency, 2006.

[MD08]   Stefan Maass and Jürgen Döllner. Seamless integration of labels into interactive virtual 3D environments using parameterized hulls. In *Proceedings of the 4th Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 33–40. Eurographics Association, 2008.

[ME11]   Patrick Moran and David Ellsworth. Visualization of AMR data with multilevel dual-mesh interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1862–1871, 2011.

[MG06]   Chohong Min and Frédéric Gibou. A second order accurate projection method for the incompressible Navier-Stokes equations on non-graded adaptive grids. *Journal of Computational Physics*, 219(2):912–929, 2006.

[MIK]   MIKE FLOOD – Toolbox for professional flood modellers. `https://www.mikepoweredbydhi.com/products/mike-flood` (last visited on January, 18th 2020).

[MN88]   Don P. Mitchell and Arun N. Netravali. Reconstruction filters in computer graphics. *ACM SIGGRAPH Computer Graphics*, 22(4):221–228, 1988.

[MRH+05]   Alan M. MacEachren, Anthony Robinson, Susan Hopper, Steven Gardner, Robert Murray, Mark Gahegan, and Elisabeth Hetzler. Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science*, 32(3):139–160, 2005.

[MWK14]   Mahsa Mirzargar, Ross Whitaker, and Robert Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2654–2663, 2014.

[Ney03]   Fabrice Neyret. Advected textures. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 147–153. Eurographics Association, 2003.

[NKF+16]   Matthias Nießner, Benjamin Keinert, Matthew Fisher, Marc Stamminger, Charles Loop, and Henry Schäfer. Real-time rendering techniques with hardware tessellation. *Computer Graphics Forum*, 35(1):113–137, 2016.

[NS13]   Mohamed Ali Nasly and Faisal Nadeem Saher. A visual aided DSS prototype development to mitigate the flood hazards. *American Journal of Geographic Information System*, 2(3):37–46, 2013.

[NSB13]   Michael B. Nielsen, Andreas Söderström, and Robert Bridson. Synthesizing waves from animated height fields. *ACM Transactions on Graphics*, 32(1):2:1–9, 2013.

[ODR⁺02]   William L. Oberkampf, Sharon M. DeLand, Brian M. Rutherford, Kathleen V. Diegert, and Kenneth F. Alvin. Error and uncertainty in modeling and simulation. *Reliability Engineering & System Safety*, 75(3):333–357, 2002.

[Off]   Offene Daten Köln. `http://www.offenedaten-koeln.de/` (last visited on January, 18th 2020).

[PA01]   Simon Premože and Michael Ashikhmin. Rendering natural waters. *Computer Graphics Forum*, 20(4):189–200, 2001.

[Par10]   Wendy S. Parker. Predicting weather and climate: Uncertainty, ensembles and probability. *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, 41(3):263–272, 2010.

[Peg]   PegelAlarm Bürgerservice, Hochwasser-Warndienst. `https://www.pegelalarm.at` (last visited on January, 18th 2020).

[PG07]   Renato Pajarola and Enrico Gobbetti. Survey of semi-regular multiresolution models for interactive terrain rendering. *The Visual Computer*, 23(8):583–605, 2007.

[PHT15]   Vsevolod Peysakhovich, Christophe Hurter, and Alexandru Telea. Attribute-driven edge bundling for general graphs with applications in trail analysis. In *8th IEEE Pacific Visualization Symposium (PacificVis)*, pages 39–46. IEEE, 2015.

[Pin]   Pinellas County – Know Your Zone. `http://kyz.pinellascounty.org` (last visited on January, 18th 2020).

[PSH⁺05]   Tim N. Palmer, Glenn J. Shutts, Renate Hagedorn, Francisco J. Doblas-Reyes, Thomas Jung, and Martin Leutbecher. Representing model uncertainty in weather and climate prediction. *Annual Review of Earth and Planetary Sciences*, 33:163–193, 2005.

[PXY⁺05]   Doantam Phan, Ling Xiao, Ron Yeh, Pat Hanrahan, and Terry Winograd. Flow map layout. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 219–224, 2005.

[Ree00]   William Gareth Rees. The accuracy of digital elevation models interpolated to higher resolutions. *International Journal of Remote Sensing*, 21(1):7–20, 2000.

[RIO]   RIOCOM – flowing competence. `http://riocom.at/en/` (last visited on January, 18th 2020).

[Riv]   RiverFlow2D – Two-dimensional combined hydraulic and hydrologic flexible-mesh model. `http://www.hydronia.com/riverflow2d/` (last visited on January, 18th 2020).

[Rou13]   Nicolas P. Rougier. Shader-based antialiased dashed stroked polylines. *Journal of Computer Graphics Techniques*, 2(2):91–107, 2013.

[RRPC12] Oscar Ripolles, Francisco Ramos, Anna Puig-Centelles, and Miguel Chover. Real-time tessellation of terrain on graphics hardware. *Computers & Geosciences*, 41:147–155, 2012.

[RWF+13] Hrvoje Ribičić, Jürgen Waser, Raphael Fuchs, Günter Blöschl, and Eduard Gröller. Visual analysis and steering of flooding simulations. *IEEE Transactions on Visualization and Computer Graphics*, 19(6):1062–1075, 2013.

[RWG+12] Hrvoje Ribičić, Jürgen Waser, Roman Gurbat, Bernhard Sadransky, and M. Eduard Gröller. Sketching uncertainty into simulations. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2255–2264, 2012.

[SATS07] David A. Stainforth, Miles R. Allen, Edward R. Tredger, and Leonard A. Smith. Confidence, uncertainty and decision-support relevance in climate predictions. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1857):2145–2161, 2007.

[SD18]   Yusuf Sermet and Ibrahim Demir. Flood Action VR: A virtual reality framework for disaster awareness and emergency response training. In *Proceedings of the International Conference on Modeling, Simulation & Visualization Methods*, pages 65–68. CSREA Press, 2018.

[SEC]   SECOM: Serious Community 2.0 Prevent Flooding. `http://nordsee.lfi.rwth-aachen.de/secom/` (last visited on January, 18th 2020).

[SH05]   Christian Sigg and Markus Hadwiger. Fast third-order texture filtering. In Matt Pharr, editor, *GPU Gems 2*, pages 313–329. Addison-Wesley, 2005.

[Sha]   Shadertoy – Adaptive grid interpolation. `http://shadertoy.com/view/WsXXRf` (last visited on January, 18th 2020).

[Sil86]   Bernard W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC Press, 1986.

[SSB+09] Alison Shaw, Stephen Sheppard, Sarah Burch, David Flanders, Arnim Wiek, Jeff Carmichael, John Robinson, and Stewart Cohen. Making local

futures tangible—synthesizing, downscaling, and visualizing climate change scenarios for participatory capacity building. *Global Environmental Change*, 19(4):447–463, 2009.

[StA13]   Jean-Francois St-Amour. Rendering Assassin's Creed III. Game Developers Conference, 2013.

[Sura]   Survey to evaluate a novel 3D flood and stormwater visualization. `https://goo.gl/forms/DqrLM6cCkroWLs9q2` (last visited on January, 18th 2020).

[Surb]   Results of the survey to evaluate a novel 3D flood and stormwater visualization. `http://drivenbynostalgia.com/files/waterrendering/survey_results.xlsx` (last visited on January, 18th 2020).

[SWH15]   Graham Sellers, Richard S. Wright, and Nicholas Haemel. *OpenGL Superbible: Comprehensive Tutorial and Reference*. Addison-Wesley, 7th edition, 2015.

[SWV$^+$11] Roeland Scheepens, Niels Willems, Huub Van de Wetering, Gennady Andrienko, Natalia Andrienko, and Jarke J. Van Wijk. Composite density maps for multivariate trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2518–2527, 2011.

[TB03]   James W. Taylor and Roberto Buizza. Using weather ensemble predictions in electricity demand forecasting. *International Journal of Forecasting*, 19(1):57–70, 2003.

[TBB09]   Natalya Tatarchuk, Joshua Barczak, and Bill Bilodeau. Programming for real-time tessellation on GPU. Technical report, AMD, Inc., 2009.

[Tes01]   Jerry Tessendorf. Simulating ocean water. *Simulating Nature: Realistic and Interactive Techniques, ACM SIGGRAPH Course #47 Notes*, pages 3:1–19, 2001.

[TJV$^+$17] Jin Teng, Anthony J. Jakeman, Jai Vaze, Barry F. W. Croke, Dushmanta Dutta, and Shaun Kim. Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environmental Modelling & Software*, 90:201–216, 2017.

[Tob87]   Waldo R. Tobler. Experiments in migration mapping by computer. *The American Cartographer*, 14(2):155–163, 1987.

[TUF]   TUFLOW – Numerical engines for simulating free-surface water flow. `http://www.tuflow.com/` (last visited on January, 18th 2020).

132

[TvW99]   Alexandru Telea and Jarke J. van Wijk. Simplified representation of vector fields. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years*, pages 35–42, 1999.

[VGB+16]   Samuel Van Ackere, Hanne Glas, Jochem Beullens, Greet Deruyter, Alain De Wulf, and Philippe De Maeyer. Development of a 3D dynamic flood WebGIS visualisation tool. *International Journal of Safety and Security Engineering*, 6(3):560–569, 2016.

[vH11]   Frans van Hoesel. Tiled directional flow. In *ACM SIGGRAPH Posters*, page 19:1. ACM, 2011.

[Vid]   Flood and stormwater modelling – Hydrodynamic modelling with VISDOM. `https://www.youtube.com/watch?v=GBP97uc7eTk` (last visited on January, 18th 2020).

[Vis]   Visdom – An integrated visualization system. `http://visdom.at` (last visited on January, 18th 2020).

[VKG04]   Ivan Viola, Armin Kanitsar, and Meister Eduard Gröller. Importance-driven volume rendering. In *Proceedings of the IEEE Conference on Visualization*, pages 139–146. IEEE, 2004.

[Vla10]   Alex Vlachos. Water flow in Portal 2. Advances in Real-Time Rendering in 3D Graphics and Games II course, ACM SIGGRAPH, 2010.

[Vul]   Visualization of Object-Centered Vulnerability to Possible Flood Hazards. `http://visdom.at/media/videos/mp4/vulnerability_camera_ready.mp4` (last visited on January, 18th 2020).

[WFR+10]   Jürgen Waser, Raphael Fuchs, Hrvoje Ribičić, Benjamin Schindler, Günter Blöschl, and Meister Eduard Gröller. World Lines. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1458–1467, 2010.

[WHM18]   Pascaline Wallemacq, Rowena House, and Denis McClean. UNISDR and CRED report: Economic losses, poverty & disasters (1998–2017). Technical report, Centre for Research on the Epidemiology of Disasters (CRED), School of Public Health, Université catholique de Louvain, 2018. `https://www.cred.be/sites/default/files/CRED_Economic_Losses_10oct.pdf` (last visited on January, 18th 2020).

[WKL+01]   Gunther H. Weber, Oliver Kreylos, Terry J. Ligocki, John M. Shalf, Hans Hagen, Bernd Hamann, Kenneth I. Joy, and Kwan-Liu Ma. High-quality volume rendering of adaptive mesh refinement data. In *Proceedings of the Vision Modeling and Visualization Conference*, pages 121–128. Aka GmbH, 2001.

[WKS+14]   Jürgen Waser, Artem Konev, Bernhard Sadransky, Zsolt Horváth, Hrvoje Ribičić, Robert Carnecky, Patrick Kluding, and Benjamin Schindler. Many Plans: Multidimensional ensembles for visual decision support in flood management. *Computer Graphics Forum*, 33(3):281–290, 2014.

[Wor96]   Steven Worley. A cellular texture basis function. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 291–294. ACM, 1996.

[WWH+00]   Manfred Weiler, Rüdiger Westermann, Chuck Hansen, Kurt Zimmermann, and Thomas Ertl. Level-of-detail volume rendering via 3D textures. In *Proceedings of the IEEE Symposium on Volume Visualization*, pages 7–13. ACM, 2000.

[YHK07]   Cem Yuksel, Donald H. House, and John Keyser. Wave particles. *ACM Transactions on Graphics*, 26(3):99:1–8, 2007.

[YNBH09]   Qizhi Yu, Fabrice Neyret, Eric Bruneton, and Nicolas Holzschuch. Scalable real-time animation of rivers. *Computer Graphics Forum*, 28(2):239–248, 2009.

Appendix A: Survey of Interactive Flood Visualization

The original, digital version of this questionnaire is available online [Sura].

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

We ask you to evaluate a novel technique for 3D flood and stormwater visualization. In this survey, you will see eight short videos in which this visualization of water is demonstrated for different scenarios and use cases in our software Visdom. In each video, please only evaluate the visualization of water.

WEITER

Gib niemals Passwörter über Google Formulare weiter.

Google Formulare

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

* Erforderlich

Personal Information

Profession: *

Meine Antwort

Areas of responsibility: *

Meine Antwort

ZURÜCK   WEITER

Gib niemals Passwörter über Google Formulare weiter.

Google Formulare

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

Question 1 / 8: Visualization of River Floods

Please evaluate the visualization of river floods in rural and urban regions presented in the video.

To watch the video in fullscreen mode on desktop computers, click the YouTube logo in the video's control bar.

How relevant is the presented visualization to your areas of responsibility? *

	1	2	3	4	5	
Not relevant	○	○	○	○	○	Very relevant

How do you assess the visualization of the presented scenarios? *

	1	2	3	4	5	
Very bad	○	○	○	○	○	Very good

Comments

Meine Antwort

ZURÜCK   WEITER

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

Question 2 / 8: Stormwater Visualization

Please evaluate the stormwater visualization in rural and urban regions presented in the video.

To watch the video in fullscreen mode on desktop computers, click the YouTube logo in the video's control bar.

How relevant is the presented visualization to your areas of responsibility? *

	1	2	3	4	5	
Not relevant	○	○	○	○	○	Very relevant

How do you assess the visualization of the presented scenarios? *

	1	2	3	4	5	
Very bad	○	○	○	○	○	Very good

Comments

Meine Antwort

ZURÜCK   WEITER

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

* Erforderlich

Question 3 / 8: Visualization of Floodwall Overtopping

Please evaluate the visualization of floods caused by floodwall overtopping presented in the video.

To watch the video in fullscreen mode on desktop computers, click the YouTube logo in the video's control bar.

How relevant is the presented visualization to your areas of responsibility? *

	1	2	3	4	5	
Not relevant	○	○	○	○	○	Very relevant

How do you assess the visualization of the presented scenario? *

	1	2	3	4	5	
Very bad	○	○	○	○	○	Very good

Comments

Meine Antwort

ZURÜCK   WEITER

Gib niemals Passwörter über Google Formulare weiter.

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

Question 4 / 8: Visualization of a Floodwall Breach

Please evaluate the visualization of floods caused by a floodwall breach presented in the video.

To watch the video in fullscreen mode on desktop computers, click the YouTube logo in the video's control bar.

How relevant is the presented visualization to your areas of responsibility? *

	1	2	3	4	5	
Not relevant	○	○	○	○	○	Very relevant

How do you assess the visualization of the presented scenario? *

	1	2	3	4	5	
Very bad	○	○	○	○	○	Very good

Comments

Meine Antwort

ZURÜCK   WEITER

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

* Erforderlich

Question 5 / 8: Visualization of Dike Breaches

Please evaluate the visualization of floods caused by dike breaches presented in the video.

To watch the video in fullscreen mode on desktop computers, click the YouTube logo in the video's control bar.

How relevant is the presented visualization to your areas of responsibility? *

	1	2	3	4	5	
Not relevant	○	○	○	○	○	Very relevant

How do you assess the visualization of the presented scenarios? *

	1	2	3	4	5	
Very bad	○	○	○	○	○	Very good

Comments

Meine Antwort

ZURÜCK   WEITER

Gib niemals Passwörter über Google Formulare weiter.

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt. Missbrauch melden - Nutzungsbedingungen

Google Formulare

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

Question 6 / 8: Visualization of Floods for Planning of Short-Term Protection Measures

Please evaluate the visualization of floods for use in an interactive planning software. In the following video, scenarios are created interactively in order to plan a short-term object protection (e.g. sandbag barriers) and to test its effectiveness.

To watch the video in fullscreen mode on desktop computers, click the YouTube logo in the video's control bar.
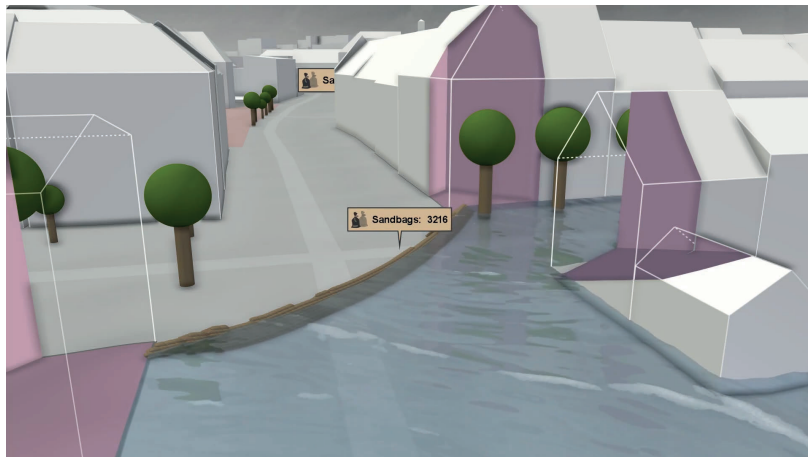
How relevant is the presented visualization to your areas of responsibility? *

	1	2	3	4	5	
Not relevant	○	○	○	○	○	Very relevant

How do you assess the visualization of floods for planning and inspection of short-term protection measures? *

	1	2	3	4	5	
Very bad	○	○	○	○	○	Very good

Comments

Meine Antwort

ZURÜCK   WEITER

Gib niemals Passwörter über Google Formulare weiter.

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

*Erforderlich

Question 7 / 8: Visualization of Floods for Planning of Long-Term Protection Measures

Please evaluate the visualization of floods for use in an interactive planning software. In the following video, scenarios are created interactively in order to plan long-term protection measures (e.g. floodwalls, retention basins) and to test their effectiveness.

To watch the video in fullscreen mode on desktop computers, click the YouTube logo in the video's control bar.

How relevant is the presented visualization to your areas of responsibility? *

	1	2	3	4	5	
Not relevant	○	○	○	○	○	Very relevant

How do you assess the visualization of floods for planning and inspection of long-term protection measures? *

	1	2	3	4	5	
Very bad	○	○	○	○	○	Very good

Comments

Meine Antwort

ZURÜCK   WEITER

Gib niemals Passwörter über Google Formulare weiter.

Survey to Evaluate a Novel 3D Flood and Stormwater Visualization

Question 8 / 8: Visualization of a Tsunami

Please evaluate the visualization of a tsunami presented in the video.

To watch the video in fullscreen mode on desktop computers, click the YouTube logo in the video's control bar.

How relevant is the presented visualization to your areas of responsibility? *

	1	2	3	4	5	
Not relevant	○	○	○	○	○	Very relevant

How do you assess the visualization of the presented scenario? *

	1	2	3	4	5	
Very bad	○	○	○	○	○	Very good

Comments

Meine Antwort

ZURÜCK   SENDEN

Appendix B: Results of Survey of Interactive Flood Visualization

An Excel table with these results is available online [Surb].

Id	1							
Profession	Universitätsprofessor							
Responsibilities	Forschung und Lehre der Hydrologie und Wasserwirtschaft							
Category	Hydrologists							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	4	5	5	5	5
Relevance	4	4	4	3	4	5	4	4

Id	2							
Profession	Wissenschaftlicher Mitarbeiter							
Responsibilities	Starkregen und Sturzfluten							
Category	Hydrologists							
Comments	**Q2**: Zusätzliche visuelle Darstellung der Wassertiefe und Fließgeschwindigkeit (eventuell in Kombination als Gefährdungsparameter für Personen) wäre wünschenswert.							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	4	5	4	4	4	5
Relevance	4	5	3	3	4	2	2	1

Id	3							
Profession	Wissenschaftlicher Mitarbeiter (Uni)							
Responsibilities	Klima/Hydrologie Forschung							
Category	Hydrologists							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	5	5	5	5	5
Relevance	4	4	2	2	1	1	1	1

Id	4							
Profession	Professor							
Responsibilities	Klimatologie, Hydrologie							
Category	Hydrologists							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	4	4	5
Relevance	4	3	4	2	4	3	4	2

Id	5							
Profession	Research Engineer							
Responsibilities	Umweltinformatik							
Category	Hydrologists							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	4	4	4	3	5	4	3
Relevance	5	5	4	3	3	2	5	2

Id	6							
Profession	wissenschaftlicher Mitarbeiter							
Responsibilities	Hydrologische und hydraulische Modellierung, Hochwasserrisikomanagement							
Category	Hydrologists							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	2	4	3	2	5	3	3
Relevance	3	3	3	3	3	3	3	1

Id	7							
Profession	Hydrologe							
Responsibilities	Modellierung							
Category	Hydrologists							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	3	4	4	4	4	4	4
Relevance	4	4	4	4	4	4	4	4

Id	8							
Profession	Hydrologe							
Responsibilities	Wasserwirtschaft							
Category	Hydrologists							
Comments	**Q1**: Die Relevanz ist abhängig von der zeitnahen Visualisierung. **Q2**: Bei der Darstellung ist jedenfalls noch eine Angabe zur Genauigkeit relevant.							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	5	4	5	5	5	5	4
Relevance	3	3	3	2	2	2	4	2

Id	9
Profession	Wissenschaftlerin
Responsibilities	Klimaanpassung, Klimaschutz, Umweltschutz, Stadterneuerung
Category	Hydrologists
Comments	**Q1**: ein Signal wäre gut, um den Beginn dynamischer Veränderungen zu signalisieren **Q2**: Sehr zurückhaltende Wasserfarbe verhindert ein Angst-Szenario, die Verfärbung der Gebäude verdeutlich die Gefahrenlage **Q3**: gute Darstellung, allerdings fehlt mir hier die Zeitachse beim Rückfluss und die Darstellung von "Resten" wie Matsch, Unrat, Müll **Q4**: Das sieht sehr gemütlich aus, sehr wenig Dramatik, die eine solche Situation ja beinhaltet. Das Wasser ist mE zu sauber und durchsichtig **Q5**: Pos: blaue Wasserfarbe verdeutlicht stärker die unterschiedlichen Tiefen, Vogelperspektive bietet ebenfalls eine gute Übersicht zur Situation. Neg: Darstellung des Deichs als hellblaue Linie wenig nachvollziehbar (Transfer schwierig von Linie zum techn. Bauwerk), ebenfalls die Verlängerung (Ausweitung Deichbruch) **Q6**: alles zu klein, vor allem die Maßnahmen könnten kurz in Lupenfunktion vorgestellt werden **Q7**: Schrägbildperspektive gefällt mir, da der menschliche Maßstab besser verdeutlicht wird, Farbgebung bei Vogelperspektive gut, Maßnahmen sind schön als Varianten angelegt und durchspielbar **Q8**: Bei der Darsellung mehr im menschlichen Maßstab bleiben (Augenhöhe), da in einer solchen Katastrophensituation die Rettung von Menschen vorrangig sein wird und nicht mehr der Objektschutz. Simulation könnte insgesamt dynamischer ablaufen,

Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	4	4	1	2	1	3	1
Relevance	2	1	2	2	1	1	1	1

Id	10
Profession	University professor
Responsibilities	Department Chair
Category	Hydrologists
Comments	**Q1**: The visualization is amazing **Q2**: Visualization is very realistic. This is exactly the tool that we need to interact with policy makers

Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5
Relevance	5	5	5	5	5	5	5	5

Id	11							
Profession	Angestellter in einem Ingenieurbüro für KTWW							
Responsibilities	Projektleiter, Projektbearbeitung im Hochwasserschutz, Risiko-management, Katastrophenschutz							
Category	Engineers							
Comments	**Q1**: Die Videodarstellung war zu klein um manche Details zu erkennen; es wäre interessant dies detaillierter darzustellen um es besser bewerten zu können							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	5	5	5	5	5	5	5
Relevance	5	5	5	5	5	5	5	1

Id	12							
Profession	Diplom Ingenieur Umweltschutztechnik							
Responsibilities	Starkregensimulation, Kanalnetzberechnung, Gewässer, GIS							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	5	5	4	5	5	3
Relevance	4	5	4	3	3	4	4	1

Id	13							
Profession	Konsulent							
Responsibilities	GF							
Category	Engineers							
Comments	**Q1**: die Zeitleiste fehlt in dieser Darstellung **Q2**: Vorwarnzeit! **Q3**: Zu jedem Hochwasserschutz sollte eine Risikoabschätzung für das Überströmen erfolgen **Q4**: Färbung der überfluteten Bereiche (Gebäude) wäre nützlich **Q6**: Bereitstellung der Schutz-materialien vorab klären **Q8**: in gefährdeten Gebieten sollte es Berechnungsmethoden zur max. Höhe der Flutwelle geben							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	5	3	4	4	4	4
Relevance	5	3	5	5	3	4	3	1

Id	14
Profession	Wasserbau-Ingenieurin
Responsibilities	Renaturierungsplanungen, Hochwasserschutzprojekte, Gewässerpflegekonzepte
Category	Engineers
Comments	**Q1**: die Farbe des Wassers ist gräulich und etwas dunkel, es könnte eine Spur heller sein (Geschmackssache) **Q4**: die Wellen und Strömungsform sind sehr gut erkennbar **Q8**: Das Wasser sieht in dem Fall zunächst wie Öl aus (als es die Häuser überströmt).

Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	4
Relevance	5	5	4	4	5	3	5	1

Id	15
Profession	Beratender Ingenieur
Responsibilities	Wasserbau Hochwasserschutz
Category	Engineers
Comments	**Q1**: interessant für Bürger, Fachleute lesen auch die Modellergebnisse ganz gut **Q2**: kräftigere Farbe wäre mM hilfreich; Färbung der GEbäude in Abhängigkeit deer Wassertiefe ist interessant **Q3**: sehr realistisch **Q5**: nicht viel anders als auch die Erebnisse von 2D modellen visualisiert werden können, was mM völlig ausreichend ist. Mehr ist für Hollywood-Filmproduktionen interessant **Q6**: sehr "bürgerfreundlich", weil leicht verständlich. Inhaltlich nicht anders als eine undurchlässige Barriere, wo die Wasserspiegelhöhe ermittelt wird. Erstellung von Szearien in einer fertigen Berechnung, wenn die Rechnung so schnell erfolgt wie die Darstellung, ist jedenfalls interessant im Einsatzfall. **Q7**: grafisch beeindruckend, wie weit die Hydraulik dabei stimmt, kann nicht beurteilt werden. Für die Bewilligung sind Wasserspüiegeldifferenzen nötig - geht das? Videos helfen bei Juristen nur bedingt. **Q8**: interessant ist das Rückbranden der Welle

Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	5	5	3	5	3	4
Relevance	4	5	3	3	5	4	4	1

Id	16							
Profession	Kulturtechniker							
Responsibilities	Gewässerrenaturierungen, Hochwasserrisikomanagment, Katastrophenschutz							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	4	5	5	4	5	5	5
Relevance	5	5	5	5	5	5	5	5

Id	17							
Profession	Dipl.-Ing. für Landschaftsplanung und Landschaftspflege							
Responsibilities	Abteilungsleiter interaktive Visualisierung und Simulation							
Category	Engineers							
Comments	**Q1**: Wasserdarstellung hilft gut bei der Erkennung der Fließrichtung, insgesamt perfekt gelungene grafische Abstraktion und Balance zwischen Detailgrad und Fokus auf die wesentliche Information! **Q4**: Sehr guter Effekt, ev. Hinweismarker kurz davor "hier passiert gleich die Bruchstelle" damit die Aufmerksamkeit dort ist und das Event nicht verpasst wird							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5
Relevance	5	5	5	5	5	3	4	2

Id	18							
Profession	Technischer Zeichner							
Responsibilities	Hochwasserschutz							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	5	5	4	5	5	4	5
Relevance	5	5	5	3	4	1	4	1

Id	19							
Profession	Ingenieur							
Responsibilities	Wasserbau							
Category	Engineers							
Comments	**Q2**: Kontrast zwischen grauem DGM und Wasser zu gering **Q3**: Kontrast Wasser und graues Gelände stärker unterscheiden **Q4**: Kontrast Wasser und graues Gelände stärker unterscheiden							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	2	3	3	4	3	3	4
Relevance	5	5	5	5	3	4	4	4

Id	20							
Profession	Vermessungstechniker							
Responsibilities	Bestandsaufnahmen und Planung von Hochwasserrückhalte-becken							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5
Relevance	3	2	1	1	3	4	3	1

Id	21							
Profession	Wasserbauingenieur							
Responsibilities	Abflusssimulation 2D und 1D, Abflussstudien, wasserbauliche Planungen							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5
Relevance	5	5	5	5	5	5	5	4

Id	22							
Profession	Projektingenieur Wasserbau							
Responsibilities	Wasserbau, hydraulische Modellierung							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	3	4	5	4
Relevance	4	2	4	3	3	4	4	1

Id	23							
Profession	Ing.							
Responsibilities	Wasserwirtschaft							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	5	4	4	4	4	5
Relevance	3	3	3	2	4	4	4	2

Id	24							
Profession	Projektingenieur							
Responsibilities	Siedlungswasserwirtschaft/ Wasserwirtschaft							
Category	Engineers							
Comments	**Q8**: Werden die nachfolgenden Wellen ebenfalls berücksichtigt oder nur die Hauptwelle?							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	5	3	5	5	3
Relevance	4	4	4	4	4	3	5	1

Id	25							
Profession	Projektingenieurin							
Responsibilities	Siedlungswasserwirtschaft, Regenwasserbehandlung, Hydraulische Berechnung							
Category	Engineers							
Comments	**Q2**: wenn die Video mit Audio ist, wäre vielleicht besser. **Q4**: Die Gebäude werden weiß gezeichnet, d.h. die Gebäude werden davon nicht beeinflusst? **Q6**: Was bedeutet die Zahländerung von dem Sandbags?							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	3	3	4	5	5	5
Relevance	4	4	4	3	4	4	4	3

Id	26							
Profession	Zivilingenieur							
Responsibilities	Wasserbau, Flussrenaturierung, Risikomanagement							
Category	Engineers							
Comments	**Q1**: Schön wäre, wenn die Fließrichtung noch besser hervorgehen würde. Vor allem auch bei Überflutungen. **Q4**: Einfärbung der betroffenen Gebäude fehlt mir hier - aber ich nehme an, das kann nachgeführt werden. **Q5**: Die Visualisierung in Radfeld ist etwas verwirrend, ansonsten sehr schön **Q6**: Genial! **Q7**: Genial!							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	5	5	4	4	5	5	5
Relevance	4	2	2	2	3	2	3	1

Id	27							
Profession	Ingenieur							
Responsibilities	Hochwasser							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	4	5	4	4	5	5	3
Relevance	4	3	3	3	4	4	5	1

Id	28							
Profession	Wasserbau							
Responsibilities	Simulation							
Category	Engineers							
Comments	**Q1**: Hübsch							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	3	3	4	4	3	3	3
Relevance	1	1	1	1	3	3	2	1

Id	29							
Profession	Planer Kulturtechnik und Wasserwirtschaft							
Responsibilities	Planung							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	2	4	4	4	3	4	4	4
Relevance	3	4	3	2	3	4	4	1

Id	30							
Profession	Diplom-Ingenieur							
Responsibilities	Hochwasserrisikomangement							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	4	5	5	5
Relevance	5	5	5	5	5	5	5	1

Id	31							
Profession	Bauingenieur							
Responsibilities	Wasserbau							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	4	5	3
Relevance	5	5	5	5	5	4	5	2

Id	32							
Profession	Kulturtechniker/Projektmitarbeiter							
Responsibilities	Wasserbau/Modellierung							
Category	Engineers							
Comments	**Q1**: Sehr gut für die Bewusstseinsbildung in der Bevölkerung geeignet.							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	5	5	5	5	5	5
Relevance	4	4	5	5	5	5	5	2

Id	33
Profession	Siedlungswasserwirtschaftler
Responsibilities	integrale Entwässerungsplanung
Category	Engineers
Comments	**Q1**: Im Film ging es weitgehend um Flusshochwasser, weniger um urbane Sturzfluten, in dem ein Tätigkeitsfeld hauptsächlich liegt, daher "eher mittel relevant" **Q2**: Diesmal passt die Darstellung voll zum Aufgabengebiet! **Q3**: Hochwasser aus Flüssen ist weniger mein Gebiet, sieht aber toll aus und zeigt sehr gut die Folgen beim "Versagen" der Schutzwand

Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5
Relevance	3	5	2	2	1	3	4	1

Id	34
Profession	Kulturtechniker und Wasserwirtschafter
Responsibilities	Hochwassermanagement
Category	Engineers
Comments	**Q2**: Das Wasser erscheint (vor allem in den Draufsichten) zu grau und ist daher leicht zu verwechseln mit einer Geländeschummerung in einem Laserscan. Die Wellenbewegung (die das fließende Wasser leichter erkennbar machen soll), ist kaum zu erkennen. Die 3D-Darstellungen sind besser, wobei auch dort das Wasser sehr grau erscheint. **Q3**: Die darstellung der Überströmung ist in der Draufsicht kaum zu erkennen. Wichtig wäre hier eine (farbliche) Unterscheidung der Überströmhöhe. **Q5**: Die Draufsichten sind gut gelöst, die 3D-Ansichten sind sehr gewöhnungsbedürftig. **Q6**: TOP!

Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	2	3	4	2	5	5	5
Relevance	5	2	5	5	5	5	5	1

Id	35
Profession	Kulturtechniker
Responsibilities	Planung
Category	Engineers

Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	5	5	5	3
Relevance	3	3	2	2	2	3	2	1

Id	36							
Profession	Projektmanager							
Responsibilities	ökologischer Wasserbau							
Category	Engineers							
Comments	**Q1**: relevant wäre Darstellung von betroffenen Gebäuden durch Aufspiegelung > 1 cm bei HQ 30, bzw. Grünland > 10 cm bei HQ 30							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	4	4	4
Relevance	5	2	4	3	3	2	2	1

Id	37							
Profession	Dipl.-Ing.							
Responsibilities	Stadtentwässerung							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	5	4	4	4	3	4	4
Relevance	2	5	3	4	3	4	4	1

Id	38							
Profession	bauing							
Responsibilities	bauing							
Category	Engineers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	2	3	3	3	3	4	3	3
Relevance	1	5	3	3	3	3	3	1

Id	39							
Profession	Stadtplaner							
Responsibilities	Stadtplanung							
Category	Flood Managers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	4	4	4
Relevance	4	4	3	1	1	4	4	1

Id	40							
Profession	Abteilungsleiter							
Responsibilities	Wasserwirtschaftliche Grundlagenplanung							
Category	Flood Managers							
Comments	**Q1**: besonders gut ist die farbige Darstellung der gefährdeten Gebäude **Q2**: auch hier gefällt mir die farbige Darstellung der gefährdeten Gebäude **Q3**: die Straßenbezeichnung und der (simulierte) Pegel sollten dargestellt werden **Q4**: die gefährdeten Gebäude (farblich), die Straßenbezeichnung und der (simulierte) Pegel sollten dargestellt werden **Q5**: wie bei Antwort 4 **Q6**: Top!							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	4	4	4	5	5	2
Relevance	5	5	5	5	4	5	5	1

Id	41							
Profession	Landesbeamter							
Responsibilities	Wasserwirtschaftliche Planung							
Category	Flood Managers							
Comments	**Q1**: Für BEL und TEL sehr interessant **Q3**: Für BEL und TEL sehr interessant							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5
Relevance	5	5	5	5	5	5	5	1

Id	42							
Profession	Landesverwaltung							
Responsibilities	Schutzwasserwirtschaft							
Category	Flood Managers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	3	4	3	5	4	4	3
Relevance	2	2	2	2	4	2	2	2

Id	43							
Profession	Sachbearbeiter							
Responsibilities	Hochwasser							
Category	Flood Managers							
Comments	**Q2**: In flacheren Bereichen entspricht die gezeigte Topographie nicht den Fließwegen, sonst eine nützliche Veranschaulichung des Prozesses. **Q3**: Anschauliches Beispiel um Restrisiko zu kommunizieren **Q4**: gelungene Visulisierung, um Versagensfall zu kommunizieren **Q6**: Es ist unklar ob die Input-Daten so genaue Aussagen, wie im Video dargestellt (Menge an Sandsäcken) zulassen. Dennoch für strategische Überlegungen sicherlich spannend							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	3	5	5	4	3	4	4
Relevance	4	4	4	4	4	4	4	1

Id	44							
Profession	Projektleiter in der Wasserbauverwaltung							
Responsibilities	Renaturierungsmaßnahmen, Flussbauverwaltung							
Category	Flood Managers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5
Relevance	4	3	4	4	4	4	4	1

Id	45							
Profession	Beamter							
Responsibilities	Wasserwirtschaftliche Planung, ASV Dienst, Förderung							
Category	Flood Managers							
Comments	**Q1**: speziell die Einfärbung der Häuser nach dem Gefährdungsgrad!! **Q4**: Färbung hat die Gefährdungslage nicht unterstützt **Q5**: Man kennt sich nicht aus, warum sind im ersten Teil des Films schon die Häuser eingestaut?							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	5	2	2	3	5	3
Relevance	5	4	5	5	5	3	5	1

Id	46							
Profession	Umweltanalytiker							
Responsibilities	Wasserwirtschaft, HWRL, Partei WRG							
Category	Flood Managers							
Comments	**Q4**: Darstellung Wasser links vom Dammbruch ist sehr schlecht							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	4	4	2	4	4	4	4
Relevance	5	4	5	5	5	5	5	1

Id	47							
Profession	Vertragsbediensteter							
Responsibilities	Betrieb und Erhaltung von HW-Anlagen und HW-Dienst							
Category	Flood Managers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	4	4	4	4	4	4	4
Relevance	3	2	5	4	4	5	3	1

Id	48
Profession	Fachbereichsleiter Operativer HWS Köln
Responsibilities	Steuerung Einsatz Operativer HWS
Category	Flood Managers
Comments	**Q1**: Eine Darstellung nur des Wasser in dem blaugrauen Farbton bietet wenig Kontrast; das Auge wird nicht sofort auf die Veränderungen aufmerksam. Mit der Einfärbung der Gebäude funktioniert es hingegen sehr gut **Q2**: Auch hier ist der geringe Kontrast eher schwierig. Zudem ist in der Geländeprofilierung z.T. nicht zu erkennen, ob es sich um Erhebungen oder Senken handelt **Q3**: Hier ist das Kontrastproblem aufgrund des stärkeren Zooms (größerer Maßstab) nicht so schwierig. Insgesamt gute Darstellung. **Q4**: Die Wellen sind bei der Erkennbarkeit hilfreich. Der Kontrast zwischen mobiler Wand und Wasser ist zu gering. Vielleicht einen tieferen Blauton für das Wasser wählen? **Q5**: Es ist auf den ersten Blick schwer zu erkennen, wo eigentlich der Deich ist. Die Verbreitung des Wassers in der Fläche ist gut erkennbar **Q6**: Hier ist eine gute Erkennbarkeit gegeben. Auch die Auswahl des Szenarios ist nachvollziehbar dargestellt. Gut ist die vorherige Darstellung des Verlaufs ohne zweite Verteidigungslinie. Hier ist auch der Kontrast zu den hellen Gebäuden ausreichend. Interessant ist die Frage, ob die monolithisch dargestellten Gebäude wirklich eine wirksame Barriere darstellen (Tordurchfahrten / Fenster). **Q7**: Gut ist der direkte Alternativenvergleich! **Q8**: Hier wäre noch eine Darstellung der Wassertiefe sinnvoll, um die Aussage zu ermöglichen, bis zu welchem Bereich die Menschen im Freien gefährdet sind.

Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	2	4	2	4	4	4	3
Relevance	3	2	5	5	2	4	2	1

Id	49
Profession	Beamter
Responsibilities	Leiter der Landeswarnzentrale
Category	Flood Managers
Comments	**Q1**: großartige Datengrundlage zur Risikobeurteilung **Q3**: mittlere Relevanz für die Steiermark **Q4**: mittlere Relevanz für die Steiermark **Q5**: keine Relevanz für die Steiermark **Q6**: für taktische Maßnahmen von besonderer Relevanz **Q8**: keine Relevanz für die Steiermark

Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5
Relevance	5	5	3	3	1	5	5	1

Id	50							
Profession	Diplom-Ingenieur Siedlungswasserwirtschaft							
Responsibilities	Abteilungsleitung Netzplanung							
Category	Flood Managers							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	4	4	5
Relevance	4	5	2	1	1	3	3	1

Id	51							
Profession	Referentin							
Responsibilities	Klimaanpassung							
Category	Flood Managers							
Comments	**Q2**: Dadurch das der Niederschlag nicht dargestellt wird, wird der Zusammenhang nicht deutlich. Nicht jedes Publikum kann dann mit der Animation etwas anfangen.							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	2	4	4	4	4	4	3
Relevance	2	4	2	2	1	3	2	1

Id	52							
Profession	Verwaltung							
Responsibilities	Hochwasserrisikomanagement							
Category	Flood Managers							
Comments	**Q2**: Skalierungsinformation wäre gut. Darstellung des Niederschlagsereignisverlaufs ist sehr gut **Q5**: Plausibilität des dargestellten Abflussbildes/Füllvorganges schwer nachvollziehbar. Visualisierung eines Dammbruchs erstmalig in dieser Form gesehen.							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	5	5	5	5	4
Relevance	5	5	5	4	4	4	5	1

Id	53							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	5	5	5

Id	54							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	3	4	4	4	4	4	4

Id	55							
Category	Concerned Citizens							
Comments	**Q1**: Die gelb/orange/rot Färbung ist hilfreicher als das Wasser an sich.							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	5	4	3	5	5	5

Id	56							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	4	4	4

Id	57							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	1

Id	58							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	4	4	4	3	4	4

Id	59							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5

Id	60							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	5	5	5	4	5	5	5

Id	61							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	3	5	4	5	4	4

Id	62							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	5	3	3	5	4	4

Id	63							
Category	Concerned Citizens							
Comments	**Q2**: Auf dem Smartphone kaum zu erkennen.							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	3	2	3	2	4	4	3	4

Id	64							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	5	5	4	5	5	5

Id	65							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	4

Id	66							
Category	Concerned Citizens							
Comments	**Q1**: In den ersten drei Videos muß man sehr genau hinsehen, um überhaupt etwas zu bemerken **Q2**: Die letzten beiden find ich sehr gut **Q3**: Die beiden letzten sind top **Q4**: Ja, so sieht es aus, wenns bricht. Alle top **Q5**: Die erste Variante find ich am besten.. Die Länge des Deichbruchs und die genaue Lage wird klar definiert							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	5	5	4	3	4	5

Id	67							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	3	4	4	4	4	4

Id	68							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	5	5	5	5	4	4

Id	69							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	3	4	5	3	4	3

Id	70							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	5	4	4

Id	71							
Category	Concerned Citizens							
Comments	**Q1**: Die schatierungen sind etwas verschwommen und Wellen Muster sind zu unnatürlich es fehlt die Scham Bildung im Bereich der braundung und die Ausrichtung müsste besser ausgearbeitet werden **Q2**: Das Wasser entsteht wie aus dem nichts es kommt komplett unnatürlich rüber **Q3**: Das Wasser kommt wie ein fingerschnippen aufeinmal über die flutmauer das kommt so in der Natur nicht vor als ersten Kot das übertreten durch Wellen Schlag vor was in der Ansicht im Film nicht zu sehen ist **Q4**: Vermisse vor Durchbruch des flutschutzes die durchfeuchtungs rinsäle der Risse im Flur Schutz und eine flutwand bricht nicht gleich 50m komplett ein sie biegt sich langsam durch bevor eine Halterung bricht **Q5**: Anspielung in etwas gemäßigter geschwindigkeit währe besser **Q7**: Höhen und Gelände können während angebracht um das flossen des Wassers besser einschätzen zu können **Q8**: Vermisse in der Animation das erst weggehen des Wassers bevor die Welle auf Land trift							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	4	4	4

Id	72							
Category	Concerned Citizens							
Comments	**Q2**: Andere Farbe wäre besser für ältere Menschen zu erkennen.							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	4	5	5	5	4	4	4

Id	73							
Category	Concerned Citizens							
Comments	**Q1**: Die Idee finde ich klasse **Q4**: Super							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5

Id	74							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	3	2	3	3	3	4

Id	75							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5

Id	76							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	4	4	4

Id	77							
Category	Concerned Citizens							
Comments	**Q1**: Eine tolle Demonstration!! **Q2**: Beeindruckend,was Technik so schaft **Q6**: Super **Q8**: Besten Dank.Super Idee							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	4	3	3	5	5	5	4

Id	78							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5

Id	79							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5

Id	80							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	2	3	3	2	3	2	2	2

Id	81							
Category	Concerned Citizens							
Comments	**Q1**: Perfekt							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	4	4	4	5	5	5	5

Id	82							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	5	5	5

Id	83							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	2	3	5	5	4	5

Id	84							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	4	3	3	4	4	5

Id	85							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	4	5	5

Id	86							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	1	5	5	5	5	5	5

Id	87							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5

Id	88							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	4	4	4	4	5	4

Id	89							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	5	4	4	4	4	5

Id	90							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5

166

Id	91							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	4	4	4	4	4	4

Id	92							
Category	Concerned Citizens							
Comments	**Q2**: Bedeutet die dunkle Einfärbung Grundwasseranstieg oder Erosionsgefährdung durch Wasser von oben? **Q4**: Der Bruch war zu plötzlich. Auf der Stadtkarte wurde nichts an zufliesendem Wasser gesehen. **Q7**: In Bergbauregionen sollte der manchmal unbekannte löchrige Untergrund berücksichtigt werden können. **Q8**: Wird mit gleicher Windstärke gerechnet oder mit unterschiedlichem Aufprall auf die Küste?							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	2	4	2	3	4	3	4

Id	93							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	4	5	5	5	4	5	3

Id	94							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5

Id	95							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	4	3	4	3	4	4	4	4

Id	96							
Category	Concerned Citizens							
Questions	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Rating	5	5	5	5	5	5	5	5

Daniel Cornel
Curriculum Vitae

Last updated January 2020

PERSONAL DETAILS

Date of birth	June 16, 1988 in Nürtingen, Germany
Address	Rüdengasse 12/11, 1030 Wien, Austria
Mail	cornel@drivenbynostalgia.com
Website	http://www.drivenbynostalgia.com

EDUCATION

Doctoral programme in Technical Sciences   2014–2020
TU Wien (formerly Vienna University of Technology)
     Thesis: *Interactive Visualization of Simulation Data for Geospatial Decision Support*

Master's programme Visual Computing   2011–2014
TU Wien
     Thesis: *Analysis of Forced Random Sampling*

Bachelor's programme Media Informatics   2007–2011
TU Wien
     Thesis: *Visionaire: Integration of 3D Characters into a 2D Adventure Engine*

General higher education entrance qualification (Abitur)   1998–2007
Freies Katholisches Gymnasium St. Meinrad Rottenburg am Neckar, Germany

PROFESSIONAL EXPERIENCE

Researcher   Since 2014
VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH
     Researcher in the *Integrated Simulations* group and rendering engine developer for *Visdom*

Software engineer   2013–2018
Self-employed
     Software development for handling of GIS data and automated database completion

Internship   2012–2013
VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH
     Integration of a CUDA raytracer into the *HILITE* framework

Teaching assistant   2012–2013
TU Wien
     Undergraduate teaching assistant for courses on introductory and advanced computer graphics

Software engineer, marginally employed 2008–2012
ISW GmbH & Co. KG
> Web development and software development for an engineering office in the field of automated generation of CAD site diagrams and the combination of CAD data with databases

ADDITIONAL EXPERIENCE

Student volunteer
Visual Computing Trends 2013, Vienna
EuroVis 2012, Vienna

Cleaner and office employee (marginally employed) 2003–2007

ADDITIONAL QUALIFICATIONS

Languages	German (mother tongue)
	English (fluent)
	Latin (qualification in Latin)
Memberships	Eurographics Association

AWARDS

Best Paper Award
Interactive Visualization of Flood and Heavy Rain Simulations, EuroVis 2019

Best Paper Award (3rd Place)
Visualization of Object-Centered Vulnerability to Possible Flood Hazards, EuroVis 2015

PUBLICATIONS

Zsolt Horváth, Andreas Buttinger-Kreuzhuber, Artem Konev, **Daniel Cornel**, Jürgen Komma, Günter Blöschl, Sebastian Noelle, and Jürgen Waser. Comparison of Fast Shallow-Water Schemes on Real-World Floods. *Journal of Hydraulic Engineering*, 146(1):05019005: 1–16, 2020.

Katharina Krösl, Harald Steinlechner, Johanna Donabauer, **Daniel Cornel**, and Jürgen Waser. Master of Disaster: Virtual-Reality Response Training in Disaster Management. *Proceedings of the 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry*, 49:1–2, 2019.

Silvana Zechmeister, **Daniel Cornel**, and Jürgen Waser. 3D Annotations for Geospatial Decision Support Systems. *Journal of WSCG*, 27(2):141–150, 2019.

Daniel Cornel, Andreas Buttinger-Kreuzhuber, Artem Konev, Zsolt Horváth, Michael Wimmer, Raimund Heidrich, and Jürgen Waser. Interactive Visualization of Flood and Heavy Rain Simulations. *Computer Graphics Forum (Proceedings EuroVis 2019)*, 38(3):25–39, 2019. **Best Paper Award**.

Jürgen Waser, Ingo Schwerdorf, Andreas Buttinger-Kreuzhuber, **Daniel Cornel**, Hendrik Schulze, and Günter Blöschl. Interaktive Simulationen als Entscheidungshilfe für wassersensible Stadtgestaltung. *Tag der Hydrologie*, Poster, 2019.

Jürgen Waser, Artem Konev, and **Daniel Cornel**. On-the-Fly Decision Support in Flood Management. *GIM International*, 32(6):22–25, 2018.

Andreas Buttinger-Kreuzhuber, Jürgen Waser, Zsolt Horváth, **Daniel Cornel**, Artem Konev, and Günter Blöschl. Outflow Boundaries for Hydrodynamic Simulations at Ungauged Locations. 9th *Annual Symposium of the Doctoral Programme on Water Resource Systems*, Poster, 2018.

Artem Konev, Manuel Matusich, Ivan Viola, Hendrik Schulze, **Daniel Cornel**, and Jürgen Waser. Fast Cutaway Visualization of Sub-Terrain Tubular Networks. *Computers & Graphics*, 75:25–35, 2018.

Daniel Cornel, Robert F. Tobler, Hiroyuki Sakai, Christian Luksch, and Michael Wimmer. Forced Random Sampling: Fast Generation of Importance-Guided Blue-Noise Samples, *The Visual Computer*, 33(6):833–843, 2017.

Daniel Cornel, Artem Konev, Bernhard Sadransky, Zsolt Horváth, Andrea Brambilla, Ivan Viola, and Jürgen Waser. Composite Flow Maps. *Computer Graphics Forum (Proceedings EuroVis 2016)*, 35(3):461–470, 2016.

Zsolt Horváth, Rui A. P. Perdigão, Jürgen Waser, **Daniel Cornel**, Artem Konev, and Günter Blöschl. Kepler Shuffle for Real-World Flood Simulations on GPUs. *The International Journal of High Performance Computing Applications*, 30(4):379–395, 2016.

Zsolt Horváth, Jürgen Waser, Andreas Buttinger-Kreuzhuber, **Daniel Cornel**, Artem Konev, and Günter Blöschl. Comparison and Validation of Three Shallow Water Schemes on Synthetic and Real-World Cases. 7th *Annual Symposium of the Doctoral Programme on Water Resource Systems*, Poster, 2016.

Daniel Cornel, Artem Konev, Bernhard Sadransky, Zsolt Horváth, Eduard Gröller, and Jürgen Waser. Visualization of Object-Centered Vulnerability to Possible Flood Hazards. *Computer Graphics Forum (Proceedings EuroVis 2015)*, 34(3):331–341, 2015. **Best Paper Award (3rd Place)**.

Zsolt Horváth, Jürgen Waser, **Daniel Cornel**, Artem Konev, and Günter Blöschl. Validation of the HWP14 Scheme for the Shallow Water Equations on Real-World Cases in Stockerau, Austria. 6th *Annual Symposium of the Doctoral Programme on Water Resource Systems*, Poster, 2015.

Artem Konev, Jürgen Waser, Bernhard Sadransky, **Daniel Cornel**, Rui A. P. Perdigão, Zsolt Horváth, and M. Eduard Gröller. Run Watchers: Automatic Simulation-Based Decision Support in Flood Management. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1873–1882, 2014.